NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

THE VALUE OF NUMERICAL FORECAST PRODUCTS IN IMPROVING TACTICAL AIR DELIVERY METHODS

by

Michael Rost

June 2007

Thesis Advisor:	Wendell Nuss
Second Reader:	Neil Rowe

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE				*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2007	**3. REPORT TYPE AND DATES COVERED** Master's Thesis		
4. TITLE AND SUBTITLE: The Value of Numerical Forecast Products in Improving Tactical Air Delivery Methods				**5. FUNDING NUMBERS**
6. AUTHOR(S) Michael Rost, LCDR USN				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000				**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A				**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

This thesis investigates the development of an agent-based system to analyze meteorological model data and generate statistics for comparison purposes. With it, it is possible to research the value and level of improvement when utilizing different levels of atmospheric-model resolution for guidance in tactical decision aids. Our agent-based system automates the comparison of model data at a location in the model field with environmental data extracted from sensor data obtained from RAOB launches. Statistics were efficiently generated for the variability of the u and v components of the wind directions, to aid in the rapid determination of the variability of model data and its effects on targeting accuracy. By addressing the interoperability and adaptability of agents, this research demonstrates the usefulness of agents to extract information to rapidly compute mission-planning accuracy.

14. SUBJECT TERMS Numerical Modeling, Agent Based System, Model Resolution, Improved Accuracy, Tactical Data Application, Automated Analysis, Data Mining			**15. NUMBER OF PAGES** 103
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

THE VALUE OF NUMERICAL FORECAST PRODUCTS IN IMPROVING TACTICAL AIR DELIVERY METHODS

Michael C. Rost
Lieutenant Commander, United States Navy
B.S., University of Colorado, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN PHYSICAL OCEANOGRAPHY

from the

NAVAL POSTGRADUATE SCHOOL
June 2007

Author: Michael C. Rost

Approved by: Dr. Wendell Nuss
Thesis Advisor

Dr. Neil Rowe
Second Reader

Dr. Phillip Durkee
Chairman, Department of Meteorology

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis investigates the development of an agent-based system to analyze meteorological model data and generate statistics for comparison purposes. With it, it is possible to research the value and level of improvement when utilizing different levels of atmospheric-model resolution for guidance in tactical decision aids. Our agent-based system automates the comparison of model data at a location in the model field with environmental data extracted from sensor data obtained from RAOB launches. Statistics were efficiently generated for the variability of the u and v components of the wind directions, to aid in the rapid determination of the variability of model data and its effects on targeting accuracy. By addressing the interoperability and adaptability of agents, this research demonstrates the usefulness of agents to extract information to rapidly compute mission-planning accuracy.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

GLOSSARY

Agent Based System – "The term 'agent' describes a software abstraction, an idea, or a concept, similar to object-oriented programming terms such as methods, functions, and objects. The concept of an agent provides a convenient and powerful way to describe a complex software entity that is capable of acting with a certain degree of autonomy in order to accomplish tasks on behalf of its user. But unlike objects, which are defined in terms of *methods* and *attributes*, an agent is defined in terms of its behavior." (Wikipedia, 2007)

Atmospheric profile - A vertical representation of the atmosphere representing the same data as contained in RAOB measurements.

Climatology - "The description and scientific study of climate. Descriptive climatology deals with the observed geographic or temporal distribution of meteorological observations over a specified period of time. Scientific climatology addresses the nature and controls of the earth's climate and the causes of climate variability and change on all timescales. The modern treatment of the nature and theory of climate, as opposed to a purely descriptive account, must deal with the dynamics of the entire atmosphere–ocean–land surface climate system, in terms of its internal interactions and its response to external factors, for example, incoming solar radiation. Applied climatology addresses the climate factors involved in a broad range of problems relating to the planning, design, operations, and other decision-making activities of climate sensitive sectors of modern society." (Press, 2007)

Forecast Valid Time – The defined moment in time when the information contained in a numerical weather forecast becomes valid. Usually this is expressed in terms of GMT.

Geopotential Height – "The height of a given point in the atmosphere in units proportional to the potential energy of unit mass (geopotential) at this height relative to sea level." (Press, 2007)

GRIB format – GRIdded Binary is a general-purpose bit-oriented data-exchange format utilized as an efficient vehicle for transmitting large volumes of gridded data to automated centers over high-speed telecommunication lines using modern protocols. (World Meteorological Organization, 2005)

Gridded Weather Fields – Numerical-model outputs such as wind, temperature, precipitation, etc. located on a geographic grid, with a specific geographic projection, and grid resolution to allow for geographic presentation of the model data.

Mission Profiles – Atmospheric wind profiles built at a specific geographic location for the purpose of mission planning.

Model Layer Wind – Wind speed and direction at a single specified layer of model output. Elevation may be measured in geopotential height or at a specific atmospheric-pressure level.

Model Resolution – The defined spacing on a geographic projection of the individual grid points of a numerical model.

Parameterization – "The representation, in a dynamic model, of physical effects in terms of admittedly oversimplified parameters, rather than realistically requiring such effects to be consequences of the dynamics of the system." (Press, 2007)

Pressure Level Coordinates – A coordinate system of numerical models that follow a constant pressure level, instead of a predetermined geographic height.

Pressure Level Wind – A wind speed and direction measured at a defined level of pressure.

RAOB - "radiosonde observation—(acronym RAOB.) An evaluation of pressure, temperature, and relative humidity data received from a balloon-borne radiosonde. The processed data are usually presented in terms of geopotential height, temperature, and dewpoint at mandatory and significant pressure levels. If the position of the radiosonde is measured to determine winds aloft, then the observation is called a rawinsonde observation." (Press, 2007)

Sigma Coordinates – A terrain-following coordinate system in which the distance between levels is not constant, but the number of levels remains constant. In dynamic models it simplifies the need to test for boundary collisions, by allowing flows to follow the terrain.

Similarity Theory – "An empirical method of finding universal relationships between variables that are made dimensionless using appropriate scaling factors. The dimensionless groups of variables are called Pi groups and are found using a dimensional analysis method known as Buckingham Pi theory. Similarity methods have proved very useful in the atmospheric boundary layer, where the complexity of turbulent processes precludes direct solution of the exact governing equations." (Press, 2007)

WGS84 – "The World Geodetic System defines a reference frame for the earth, for use in geodesy and navigation. The latest revision is WGS 84 dating from 1984 (last revised in 2004), which will be valid up to about 2010." (Wikipedia, 2007)

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I want to thank all those who took the time to answer my questions, entertain my ideas, and let me forge my path, as wandering as it may have seemed. I hope the end result was well worth the struggle. The process I know held true to lending value to, if for nothing else, the lessons learned from overcoming many obstacles over the years while still maintaining focus. For that I would also like to thank all those who extended me the courtesy of ‘patience’ during this entire process.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

By bringing computational simulations into the military planning process, tactical decision aids have become critical parts of the command-and-control process. One of the key steps in every military campaign plan is determining the environmental influences on the battlespace and incorporating these various types of influences into the tactical decision aids.

When planning for future meteorological environments, several resources are available depending on the time frame for planning. The use of climatology is the primary mechanism utilized for long term planning purposes. For medium-range forecasts, usually a global model with moderate spatial and temporal resolution is used. For mission-specific short-term planning, usually the highest-resolution short-term forecast is desired. Probably the most critical issue at stake in the planning process is managing data availability against resources; because while it is always desirable to have the newest and best equipment that money can buy, the most cost-effective way to do business is to maximize available resources. This makes it necessary to balance the requirements for high-resolution forecast models with other available resources.

The problem most often encountered is to identify which resource will provide the best balance for tactical decision aids. It is necessary to identify through a sensitivity study what changes in resolution do to the output of the aids. An effective study requires a wide spectrum of data to be analyzed to judge which resolution level is most effective. To do this efficiently, automation must be used to collect, analyze and interpret the data.

This thesis explores the idea of using an agent-based system to automate this process. The agents collect and archive multiple sets of data from the varying degrees of model resolution and climatology fields. Model data was collected over the two areas in the U.S. covering Roanoke, VA and Vandenberg AFB, CA for two one-week periods during the winter and spring. The desired result is to show the improved forecasting

value of using real weather instead of long-term climatology. Sounding data from these locations was collected to provide the true wind situation and for post-analysis comparison.

Interoperability in data comparison is no simple undertaking. Even though data standards exist, they are usually applicable to a single type of data. Coincidentally we are interested in comparing and contrasting multiple types of data, each with separate standards, and no common interfaces between them. Most of the standard data types utilized today were created long before XML was brought into common use. Additionally there are many assumptions that need to be made about data scheme, including in this case the need to accurately geo-locate the data on a world grid. Even with the capability to utilize XML schemes to bring commonality to the data, there are few data bases out there with this capability, and even those presently incorporating XML schema are do so with a limited set of data.

To support the Tactical Aircraft Mission Planning System (TAMPS) tactical decision aid, the agent-based system then generated trajectories for payloads dropped from various predefined mission profiles using the RAOB launch sites at Roanoke and Vandenberg as the target destinations. The agent-based system loaded into the analysis program the specified model data of differing resolutions, and then compared the predicted data to environmental data extracted from RAOB launches. Statistics on the variability of the u and v components of the wind was collected to determine the variability of model data and its effects on targeting accuracy. The agent-based system also addressed the interoperability of statistics from the weather files for use by TAMPS for mission planning with the true wind statistics obtained from atmospheric soundings. It ingests gridded weather data in GRIB format and uses that gridded information when comparing model winds to real winds. The agent-based system makes it possible to maximize the effectiveness of a payload dropped from the release point and guided by environmental winds to a destination.

The remaining chapters of this thesis will go in-depth to describe the guiding principles of my research process, to annotate the difficulties I encountered in completing this project, to outline the results I uncovered, and to make further recommendations on future research opportunities.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. METEOROLOGICAL EFFECTS

The earth’s surface where we spend a majority of our lives is the bottom or boundary layer of the earth's atmosphere.

> We can define the boundary layer as that part of the troposphere that is directly influenced by the presence of the earth’s surface, and responds to surface forcings with a timescale of about an hour or less. These forcings include frictional drag, evaporation and transpiration, heat transfer, pollution emission, and terrain induced flow modification. The boundary layer thickness is quite variable in time and space ranging from hundreds of meters to a few kilometers. (Stull, 1988)

From the perspective of naval oceanography, it is important to accurately forecast the boundary-layer processes as they affect daily weather patterns. With accurate forecasts, we can predict important weather events, such as when clouds may clear, so we can effectively use reconnaissance assets to our tactical advantage.

However, because meteorological and oceanographic models are only approximations of how the atmosphere behaves, they are subject to error. How a model approximates the variability of the atmosphere (with its parameterization scheme) along with its spatial and time resolution strongly affects its ability to accurately forecast atmospheric changes.

Atmospheric models typically track six parameters at the grid points of each model layer. Pressure, moisture content in the form of the mixing ratio, temperature, and the three dimensional velocities are utilized to track the state of the atmosphere. The most simplified way of explaining this modeling process, is that an atmospheric model is a complex, density driven fluid dynamic problem, solved in four dimensions. The fluxes that define the work input, work derived, thermal energy input and thermal energy output of every basic thermodynamic problem are provided by model parameterizations.

Similarity theory, a method of empirically finding relationships between variables, is an important part of that model parameterization. It helps define how accurate the model's physical approximations are, and additionally provides for a

methodology to extract additional forecasting parameters such as precipitation and visibility from the six internal variables tracked by the model. Similarity theory was also utilized in this research to define the wind profiles that are extracted from each model. Additional discussion about modeling, parameterization, and the importance of the boundary layer processes are discussed in Appendix 1 of this thesis.

B. MODEL RESOLUTION

Besides the details of how a model handles the boundary layer and its free-atmosphere physics, when computational models are used to forecast weather events, the temporal and spatial resolutions of a model are key factors influencing how well the model can forecast different scales of meteorological phenomena.

The grid spacing of the model, or its spatial resolution, is important for the recognition and prognosis of boundary-layer features such as moisture transport and local winds. As model resolution increases from global scales to mesoscale, and down to microscale, the computation time increases on the order of n^2 where n is the number of model grid points of the traditional square meteorological model grids. Part of this thesis examines the effects of increasing the spatial resolution to see how much the planning aid is sensitive to the change in resolution.

The time spacing of the model, or its temporal resolution, is another key factor in the prognosis of weather features. Although some linear interpolation can be used between successive model outputs representing different times (to assist in forecasting the timing of an approaching front), few natural phenomena are linear in time. With a model of sufficient temporal resolution, it becomes easier to gauge the timing of events, at the expense of increasing the amount of data that must be transferred and analyzed and the time to do it.

III. IMPLEMENTATION

A. TEST DATA

Since the parameterization scheme in a model can strongly influence its output and because I wanted to limit my scope to resolution changes, I had to ensure that multiple resolutions of the same model were available. Initially I chose the Navy's NOGAPS and COAMPS models. After some investigation though, I discovered there are some variations in the internal physics of the model. The COAMPS model not only had finer grid resolution, but also utilized different parameterization schemes than the NOGAPS models that affect the way it forecasts some model parameters. Instead of devising ways to determine if that parameterization differences mattered to the output I instead chose use the North American Model (NAM) of the National Center for Environmental Prediction (NCEP). This model is available with several different resolutions all with the same internal parameterization to the model physics.

Data sets were downloaded from NCEP's public FTP server every twelve hours for two periods of seven days. One period was chose to provide a sample of wintertime conditions, and the other sample was for late spring. The forecast sets were all part of the North American Model for the region covering the North American Continent as shown by the NCEP NAM model website. Each of the sets were downloaded to cover twelve-hour forecast increments out to 72 hours, and for grid resolutions of 81, 32, and 12km even though some of these resolution sets were also available in 6 hour increments. For the sake of maintaining consistent comparisons, every set was available in 12-hour sets. Additionally each individual resolution set has a different set of parameters stored internally in the data, even though the data all originates from a single high-resolution model. In the case of the NAM model, NCEP chooses to run a single high resolution model and then sub-sample that data lower resolution rates. By doing sub-sampling and reducing the number of products available in the data sets they may custom tailor the file sizes delivered and their approximate download times, at the expense of degrading the available data resolution during the sub-sampling.

Further data management is accomplished by NCEP with each file having its contents stored in GRIB format (a gridded binary format with compression) allowing each individual file to range from ten to nearly one hundred megabytes total. The GRIB standard was defined by the WMO specifically for sending this type of meteorological data, and a full definition of the GRIB standard can be found on the WMO web site. Even with compression, the data volumes were quite large, and each twelve-hour period was nearly a giga-byte of data, which took approximate two and a half hours over a high-speed circuit to download. As an individual file can contain several hundred different parameters, for brevity sake a full definition of the parameters is not listed here. Full definitions of the contents of each file can be found on the NCEP public FTP server.

B. DATA MANAGEMENT FOR AGENTS

Obtaining a large amount of data every twelve hours can require some significant human interaction time. A key advantage of using agents is to reduce the amount of this time by putting some intelligence into the agent to enable it to accomplish the task autonomously. One challenge was that file paths to the data routinely changed as new forecasts were created. For this I developed an FTP agent that could predict the new file path and reliably download the data set without my intervention. Another agent created a special index for the data by forecast time so that a second agent could develop atmospheric profiles. Using these two agents alone generated substantial time savings.

Collecting large volumes of data and organizing it for statistical analysis requires good planning. Because the data sets were already in digital formats as output from a numerical model, it made sense to develop a semi-intelligent automated process to gather what was needed. The main technical challenge was to add intelligence to the automated collection program to understand the changing naming scheme used by the NCEP data servers.

Because of the large volume of data generated by the numerical model, its output data sets are only available on the server for a period of a day. In addition, the directory structure changes every day while the file names remain the same. To address this, the download agent needs built-in intelligence to understand the directory structure based on

the date, and the storage scheme had to sort the data by creating a similar structure on the local machine to hold the local copies. In the practical case of this agent individual files were downloaded via separate processes to allow data recovery in the case of connection time outs. Those data files were, upon validation, then stored with a directory structure that was labeled so as to eliminate the confusion of having common overwriting file names.

C. AGENT PROCESSING

We built a profiling agent that operated on the data sets. This agent creates a point profile of the atmosphere from the numerical forecast data. This agent analyzed the data files inside the directory to establish the model resolution, the forecast date, the forecast valid time, and the elevation level of every available wind grid field. The information on the available data (the metadata) is completely contained inside the data set as a GRIB header which is mandated for all GRIB files as part of the GRIB standard. From that information, the agent made an interpolation inside the grids of what the winds would be from the surface to the five thousand meter level at one-meter increments above to the weather observation stations for Roanoke, Virginia and Vandenberg Field, California.

The profile was then interpolated between model levels at one-meter increments so that a 5001-point profile of winds over a weather station was created; it was then written out to a "profile directory". Every file was named according to its location inside the grid, with RNK for Roanoke and VBG for Vandenberg. In addition there was a DTG identifier for the time of the forecast, and a forecast time group inside the title. With the profiles named according to this method, statistics could be generated for two comparison groups, according to both future forecast times and forecast valid times.

Then archive wind data of the measured wind profiles for Vandenberg and Roanoke, which were downloaded from the University of Wyoming Atmospheric Sciences web server, were collected manually via a GUI interface that allowed block downloads for the appropriate dates corresponding to every valid forecast time available in the profile directory. As it is standard practice in Meteorology to develop a baseline

for predictability by utilizing climatology data, additional monthly mean wind profiles for the past two years were generated manually and added to the data sets for the purpose of generating a climatological base-line for comparison purposes. With these actual measured wind data sets available, similar interpolation was done so that the matching 5001-point data set could be compared.

Next, the statistics agents compared the actual RAOB winds with the model profiles by calculating the mean error over the entire profile. These statistics were generated for the valid forecast time comparison, and also by the forecast hour, making necessary two agents with slightly different rule sets. Another agent built the chart data for inclusion in this report.

The next agent was the most time-consuming to create and simulated dropping an object through the atmosphere utilizing the wind profile sets. Once the simulations were completed for both object types, and the wind drift profiles were saved, a second type of statistics agent analyzed the drift profiles. These drift-statistics agents were modifications of the previous statistical agents, intended to collect the similar error profiles on down-range and cross-range windage errors. With the real world winds acting as the base drift profile, the forecast wind drift sets were compared to examine mean errors by forecast time and valid time. In addition, the real-world wind drifts were compared to climatology sets to see if that data was comparably accurate to the forecast data.

IV. MODELING METHODS

Most of us have driven down the road and have felt the effects that an unexpected gust of wind can have on an automobile. Even something massive like a delivery truck can be pushed about because of the large surface area it exposes to the wind. As was mentioned in the last chapter, these gusts are often found in the boundary layer of the atmosphere, but their short duration is not accounted for in our numerical models.

It is then possible that as we use these models for the purpose of planning military actions, we introduce errors into that planning cycle. The idea of my research is to investigate the planning cycle and find out how sensitive it is these errors. Practicality demanded that the scope of the project be limited though, so I chose to examine the effects of winds on target planning.

A. TDA AVAILABILITY & LIMITATIONS

Tactical decision aids (TDAs) are a common method of computer assisted planning tools for military planning applications. They give planners the benefit of being able to modify planning variables and examine the effects in order to optimize the effects of available resources.

As I began to research planning aids available that would work for the experiment, the first package I was told about was TAWS. TAWS (Target Acquisition Weapons Software) is a TDA for electro-optic effects on military systems. One of the developers at NRL Monterey, Dr. Andreas Goroch, demonstrated for me the TDA's ability to predict the effects of weather parameters on electro-optic systems such as target acquisition, and distinction. This was not the type of application I needed to complete my project, but he generously recommended that I look into another TDA program called PMPT, and a project manager at the Naval Air Systems command, Mr. Jerry Wyant, confirmed for me that PMPT (Paveway Munitions Planning Tool) was used for mission planning involving the dropping of Paveway Munitions (guided bombs) from Navy aircraft. He also gave me the name of an NPS professor Dr. Morris Driels.

I contacted Dr. Driels, who sent me literature on the trajectory methods employed by PMPT. After studying the documentation it appeared that this was the software I needed for part of my research, so I arranged a demonstration with Dr. Driels, to see how weather parameters were input into the model and how they affected the trajectory. During that demonstration, with the assistance of Dr. Driels, we concluded that the planning package did not take into account the effects of crosswinds on the falling body. With this discovery, and the fact that my research had yet to turn up a planning package that could be used for the simulation of cargo drops, I found that my research would necessitate developing my own drag models for simulations.

B. BODIES CHOSEN

As I began to research the applications available to model the dynamics of drag, it became apparent that the complexities of this project could quickly exceed the available resources. There is a large field of fluid dynamics dedicated to examining flow fields in detail far beyond a feasible limit for this research project, which would need supercomputing resources.

As I looked through the literature, I discovered two texts of particular interest for their applicability to this problem. I utilized Kuethe & Chow's text on the *Foundations of Aerodynamics 5th edition* to reference the calculations necessary to calculate drag for high speed and low speed subsonic flows. I also found Blevins text, *Applied Fluid Dynamics Handbook*, useful for gathering the needed constants for coefficients of drag for simplified bodies in a flow. After careful studies of the formulas in the texts, I developed two simplified models that closely simulate bodies of military interest to use as the test subjects in my research.

After careful studies of the formulas in the text, I settled on two simplified bodies of military interest to use as the test subjects in my research. The first body was a model of the mark 82, 500lb-unguided bomb. From online research, I found an unclassified source of information at www.Janes.com for conventional operating parameters, and designs statistics of this bomb. Utilizing the information from the previously mentioned

texts I developed the following formula to model the drag flow of this body for flow both along the axis of the body, and across the axis of the body. (Figure 1)

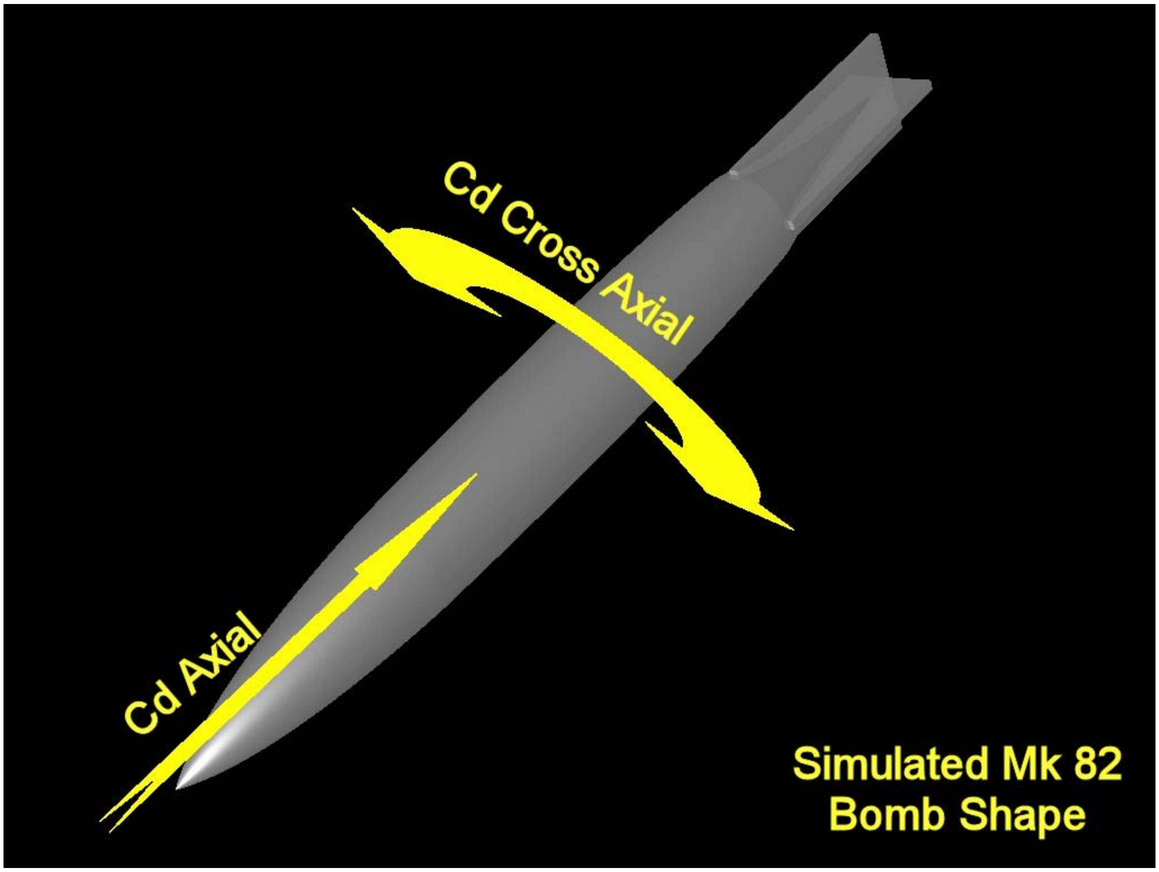


Figure 1. Drag Flow Example for Mk 82 Bomb Model

Then I developed a code set to test that my formulas worked reasonably well. My first model assumed that the bomb was dropped through the atmosphere with only gravitational effects. Figure 2 demonstrates that the results were reasonable as the bomb fell on a parabolic path as opposed to a straight line drop that would have occurred if gravity were neglected.

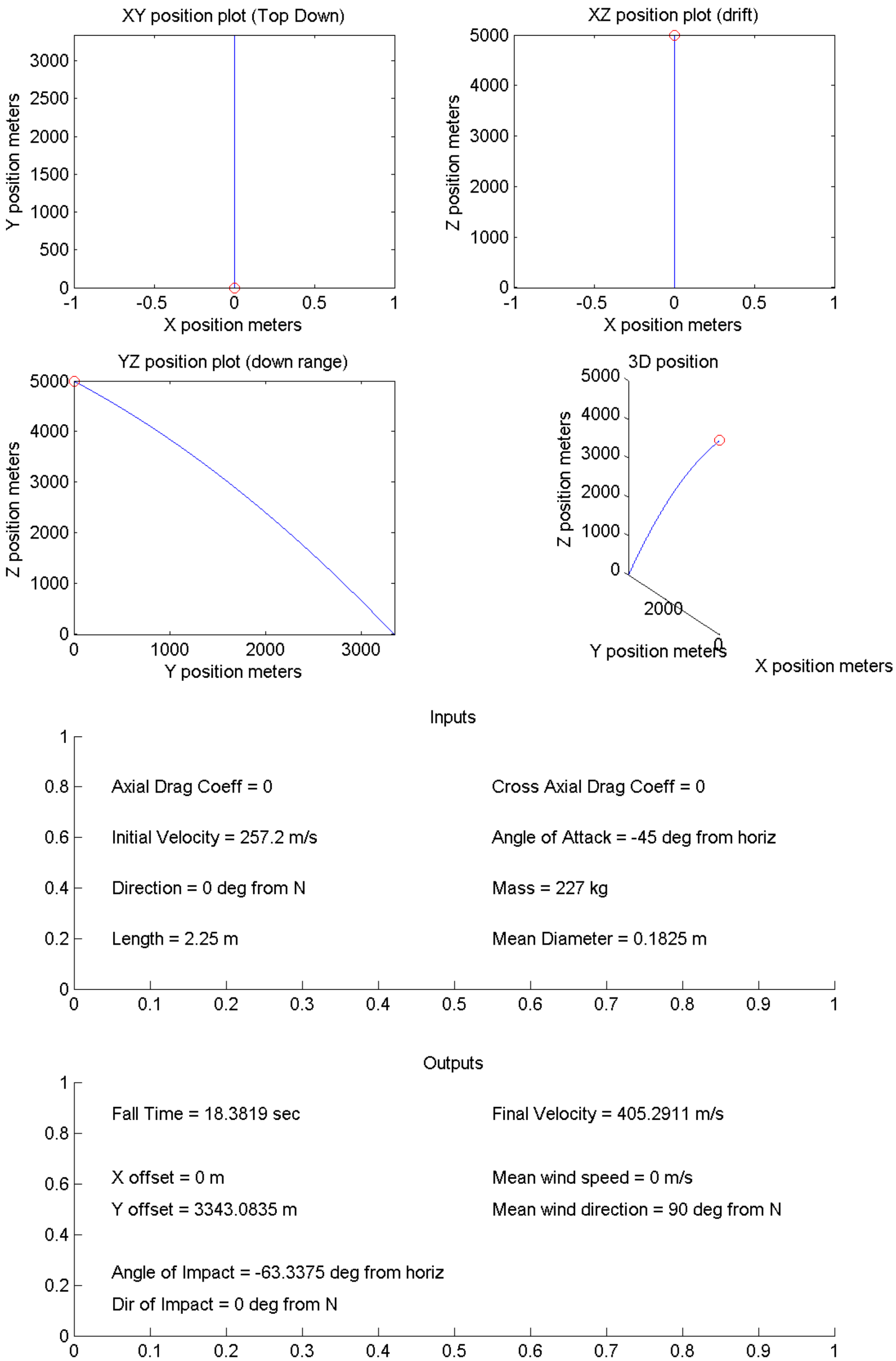


Figure 2. Bomb Model Output with Gravity Effects

Next, I made changes and adjustments to add drag effects, first along the axis of the body. That code also showed that the drag effect was reasonable, as now the bomb velocity slowed more and fell shorter in range than had the previous models results indicated. (Figure 3) With that effect working correctly, I next set to work inserting the effects of a constant velocity head wind blowing on the body as it fell through the atmosphere. With slight modifications to my calculations to change the apparent air velocity over the body to reflect the wind effects, another run was completed and showed reasonable results. (Figure 4)

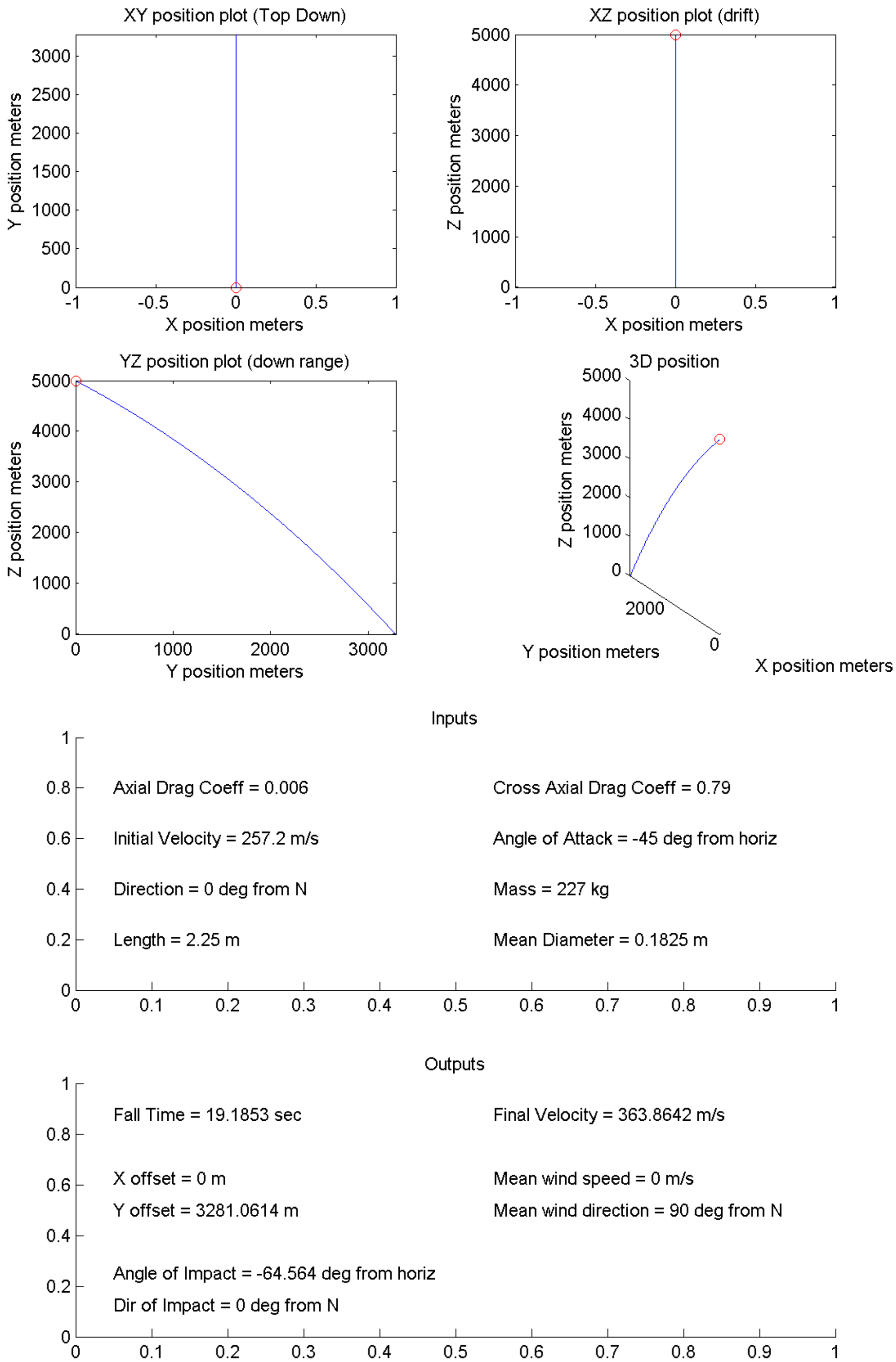


Figure 3. Bomb Model Output with Drag Effect

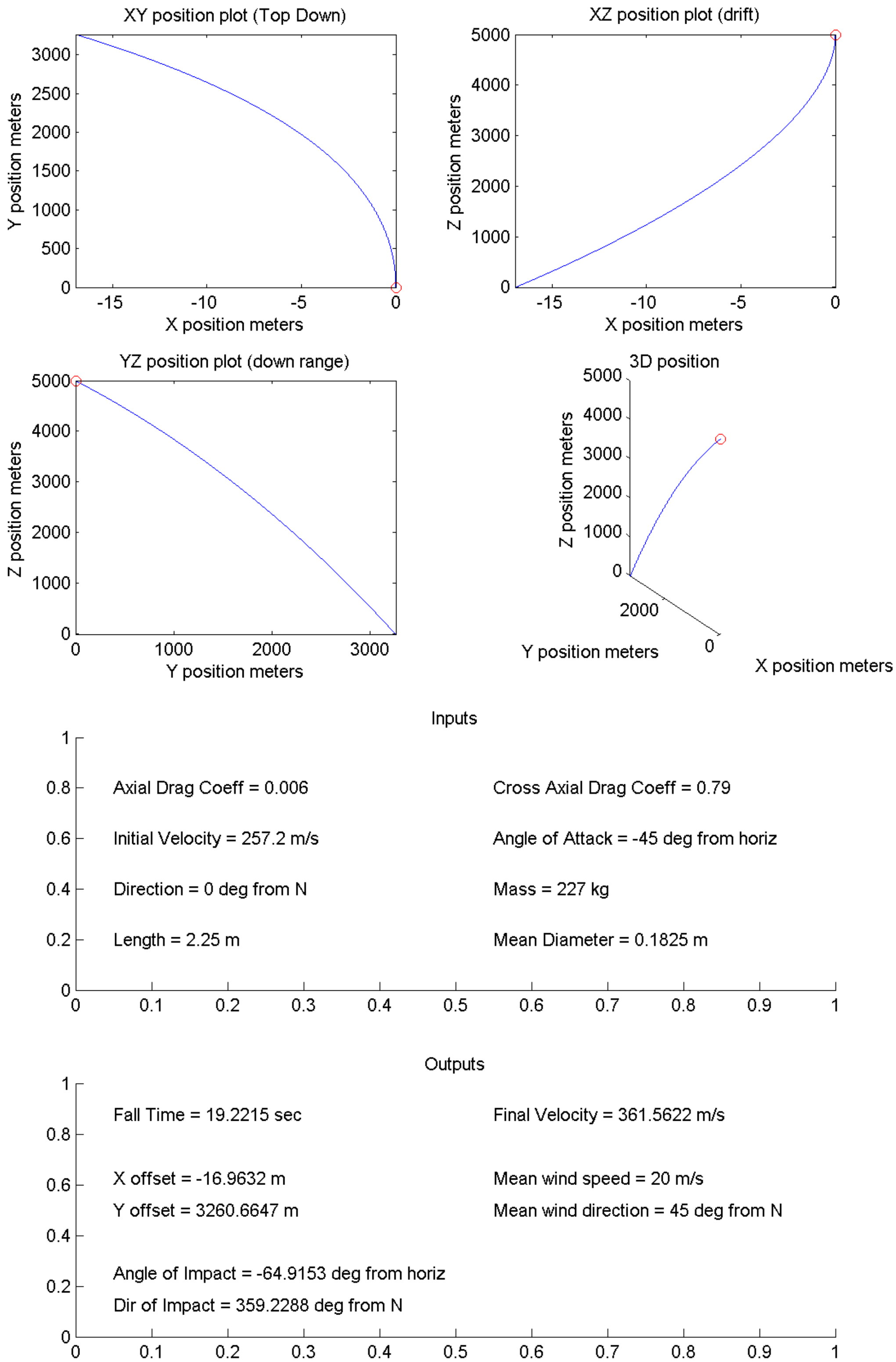


Figure 4. Bomb Model Output with Head Wind Effects

Finally, I was able to add with some difficulty the effects of crosswinds. The crosswinds used a slightly different set of drag calculations that complicated my previous equations, with another degree of variability. It took some time, including a rewrite of previous code to arrange the order of calculation correctly, but all code finally worked and showed reasonable results. As shown in Figure 4, the crosswind calculations actually push the weapon off target in the crosswind directions.

With a working model of the mark 82 bomb established, I went back to the reference texts and developed another set of equations for a parachute rigged 1000lb cubic cargo palette. For this model I made up two separate drag elements that were tethered together and then modified my previous model accordingly (Figure 5). After another trial run (Figure 6), it appeared that this model was also working correctly and work could continue to test these models with real world wind data.

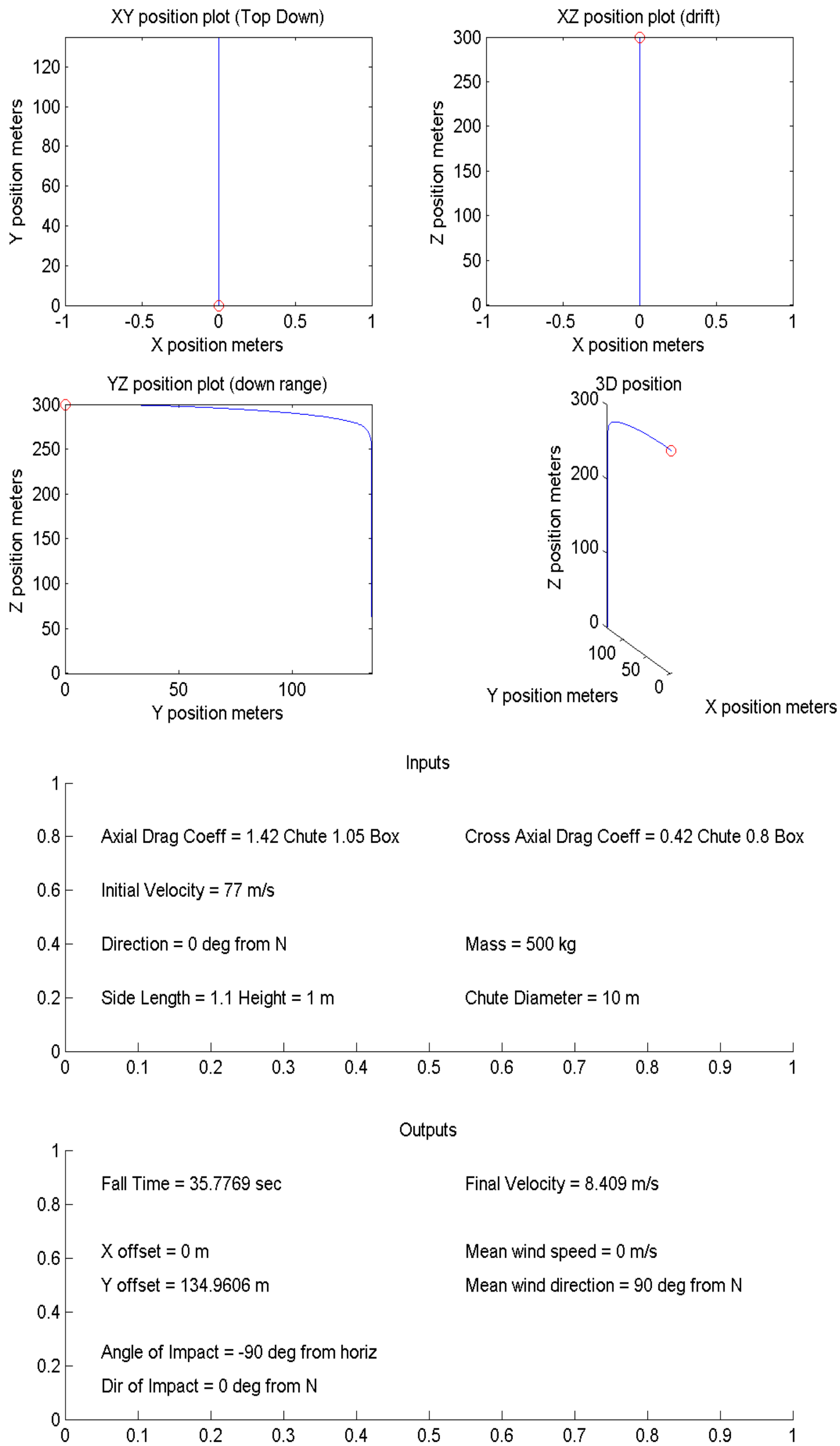


Figure 5. Chute Agent Output with Axial Drag Effects

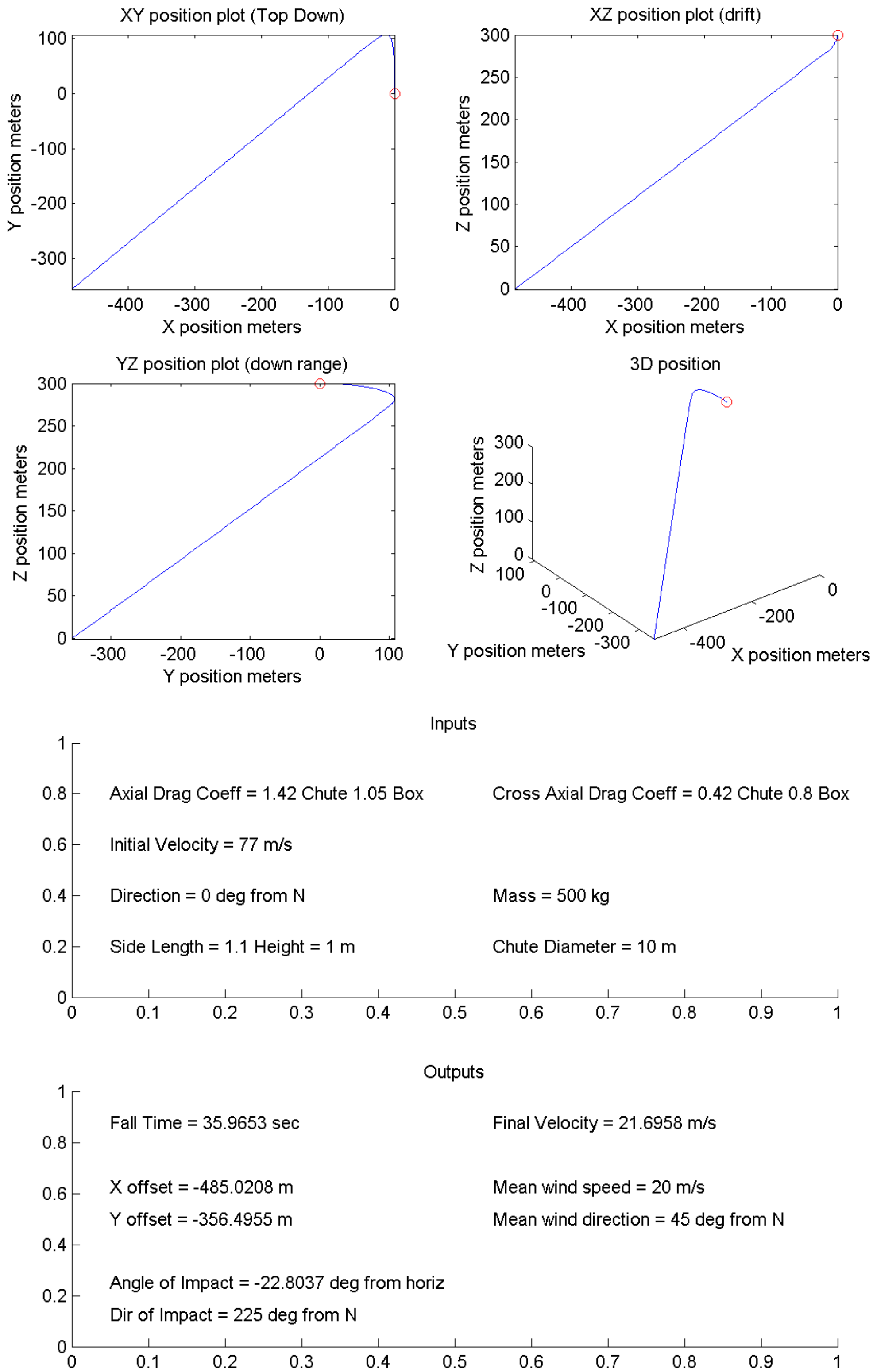


Figure 6. Chute Agent Output with Axial and Cross Axial Drag Effects

C. DATA AVAILABILITY

As was mentioned in Chapter III, because of the internal model differences used by the FNMOC regional and global models, I chose to go with the NCEP NAM model instead. The data sets were collected to coincide with two different simulated target locations where regular launches of weather balloons take place.

One of the chosen regions was Vandenberg Air Force Base (34 deg. 39min. N, 120 deg. 34 min. W) in California. This location was chosen because the upstream meteorological conditions for weather patterns are usually out over the Pacific Ocean, whereas observational data for input to the meteorological model is usually sparse in comparison to other locations. The idea behind choosing this location was to gauge model performance in comparison to other regions where input data is more plentiful.

The second location needed to be somewhat consistent with the Vandenberg location, but in a region where it had a data rich upstream source. For that purpose a location was chosen near the east coast of the United States around the same latitude, one that also made regular reports of RAOB soundings. This was Roanoke, Virginia (37 deg. 19 min. N, 79 deg. 58 min. W).

Data sets were collected for one-week periods for two different seasons of the year. The periods covered are for February 14 00Z to February 18 00Z, 2005 and April 4 00Z to April 8 12Z, 2005. All data was taken from NCEP's public FTP server, by an FTP agent that collected and sorted the data. In addition I used the public server from the University of Wyoming to collect the RAOB reports for the periods from the beginning of the data collection though 72 hours past the final date. By collecting past the last date I had a set of true wind values that correspond to the 72-hour forecast of the last model run.

After the collection period I did discover that the data sets from the NCEP server are not exactly consistent with each other over the different resolutions. Data from the 81-km and 32-km resolution sets have a set of pressure layer winds available at levels in 50mb intervals from 1000mb and up. I chose to utilize the pressure level data only as far as 500mb, which would typically represent levels near 5km above ground level. In

addition, those sets have geopotential height values for each of the pressure levels so that an altitude-to-wind speed profile can be generated over each of the respective targets. The 32-km resolution sets have additional model-layer wind data sets that can be used to supplement and fill in data in the profiles between the pressure level winds, and this data set also had wind levels at 25mb increments to further enhance the profiles.

However, in the 12-km resolution sets I found that the wind values given are a layer wind corresponding to pressure layers above ground level. This data has a form much different in structure than those of the other sets and required some corrections to be made to place it into a form similar to the other wind profiles.

Using the calculations from Chapter 2 of Kuethe and Chow's text, I found it possible to extract a density altitude correlation between the top and bottom of the layer. It made it necessary to generate a second profiling agent though to do this step. This new agent had to extract from the data sets values for the mean sea-level pressure at the profile locations. Temperature and moisture corrections were extracted, and used to calculate the density of the air parcels in this layer, and a corresponding altitude for the top and the bottom of the layer. Next, since it was a mean layer wind, the value of the winds at the middle of the layer was assumed to be exactly this extracted value, and a linear extrapolation was used between the values similar to the way the profiles are created with the pressure-level winds. A more extensive examination of these profiles will be made in the chapter on statistics and findings to see how this treatment affected the wind profiles as compared to real-world winds.

D. ASSUMPTIONS

As with all modeling and simulation, it is impossible to build a perfect model. As explained in Chapter II, even the meteorological model used for the simulation purpose comes with some errors. The purpose of this project was to minimize the amount that modeling errors affected the outcome of the project.

The drag model was built with the assumption that even at high rates of flow speed, the viscous drag effects were the primary influence on the body and turbulence was minimized. Shear effects on the body were considered minimal in effect because of

the compact size and great mass of the bodies involved. The drag coefficients used are only close approximations to those of real-world bodies. The numbers are derived from the texts of Keuthe and Chow and Blevins and could vary greatly from real world numbers, although at first estimate they appear to be well approximated to within about +/- 5% of real world values. I have assumed that wind flow is only in the horizontal plane, and all vertical wind velocities though available in the data sets are neglected. For the drop model of the palette, I chose to make an assumption that the parachute would take time to deploy as the cargo left the aircraft, so the drag effects of the parachute are assumed to increase linearly over a time of 3 seconds. This initial time is an estimate after watching video analysis of this evolution of the amount of time it would take for the chute to fully deploy.

Additional parameters for the airdrop velocity, altitude, and incline of the Mark-82 bomb are taken from the on line source "Fire Support Coordination," found at a military interest site www.gruntonline.com. While I cannot specifically attest to the validity of this data, without getting into classification concerns the parameters do appear reasonable for real world applications, and it seemed they would work well for simulation purposes.

THIS PAGE INTENTIONALLY LEFT BLANK

V. METEOROLOGICAL EFFECTS ON TDA'S

As previously mentioned, knowing and understanding the sensitivities a TDA has to a forecast parameter is paramount in understanding how well that TDA works, both from the forecasters' perspective and that of the warfighter depending on this output to make their jobs, safer and more effective. Knowing how the type and quality of the data used in simulation inputs might affect the output can give us a level of confidence helping us demonstrate how well we should or should not trust our decision-aid recommendations. To do this usually requires a great deal of data collection and analysis, especially when dealing with chaotic inputs like weather.

The intent in this research was to demonstrate how automating the process, and utilizing the advantages of ever of increased computational speeds, would make it possible to use agent-based systems to deal with demands of long term data collections and analysis. The repetitive tasks of collecting, sorting, analyzing, and displaying outcomes could be written into modular and quantifiable steps of code that could do a majority of the work for a forecaster, researcher, or scientist, thereby making it less time consuming to do research into the sensitivity of these systems to weather inputs.

Still the researcher must also have a working knowledge of the physical representations of the data they are working with to understand the relationship between the statistical sensitivities and the actual physical effects they represent.

A. THE BOUNDARY LAYER

One of the most popular teaching references for gaining insight into the complexities of the Earth's boundary layer is Stull's book "I*ntroduction to Boundary Layer Meteorology*". In the first chapter of this book Stull provides a concise explanation of the chaotic processes that take place within the boundary layer.

According to Stull, the boundary layer is a region that extends to between one and three kilometers above the Earth's surface. This is the region where a majority of life process on the planet takes place. This is also where a large portion of the fluxes that drive our atmosphere comes from. It is in the boundary layer where we introduce heat

energy into the atmosphere, both by converting light to radiant energy at the surface, and by storing that radiant energy for release after the sunsets. It is also here that moisture is introduced into the atmosphere, from many different sources ranging from obvious sources like oceans, bay, lakes and rivers, to less obvious sources like soil evaporation, snow sublimation, plant transpiration, and factory exhausts.

Stull's text also teaches that these two fluxes heat and moisture play a critical role in changing the density of the atmosphere and in causing instability that drives vertical motion. In addition the majority of atmospheric turbulence takes place in the boundary layer, playing an important role to help mix the layer and re-distribute the heat and moisture through out the layer.

Stull also goes on to explain about the effects that geography plays in the boundary layer. Spatial variation of terrain not only aids in generating additional turbulence and vertical motion, but it also plays a role in varying both heat and moisture fluxes. Specifically he makes examples of things like a sea breeze to show differential heating effects, and how something like a high mountain in tropical latitudes can affect the rainfall and moisture availability on both sides of the terrain.

It becomes quickly apparent how difficult it can be to create a model to accurately represent the boundary layer interactions. Since there is only one real-time one-to-one scale model of the Earth and we are living on it, were forced to come up with some best-fit approximations of how the Earth's complex processes work. This parameterization tries to accurately represent how moisture and energy are distributed within the boundary layer, as well as the rest of the atmosphere, so that we can numerically forecast these interactions forward in time.

Since they are only approximations of what is happening inside the box of the model that bounds them, they are subject to errors. Additionally we start with an analysis that does not have complete state information of the condition of the atmosphere for every place on the planet as well. Over time as we move the forecasts forward the errors compound reducing the overall accuracy of the forecast.

It can be deduced then that a model parameterization strongly affects its forecast performance. A model that has a well-tuned scheme that closely approximates the environment it represents should have less long-term error. Also one that has a higher resolution should better reflect the surrounding environment and also have a smaller error rate.

In this study, the scope of the model is limited to a single type, with multiple resolutions. As mentioned earlier each resolution is sub-sampled from a single high-resolution model with its own parameterization scheme, tuned to its specific resolution that may affect its long-term accuracy rate. A higher-resolution model should maintain a lower error rate over a longer period of forecast times, when compared to actual measured winds. From that information we should be able to extract, via agent modeling, information about whether that long-term error rate has a negative or neutral effect on a TDA used to plot the path of a falling object such as a bomb or a parachute retarded supply palette, and also deduce if the sub-sampling of the resolution has an effect on those trends as well.

B. HYBRID COORDINATES

Inside numerical model space a coordinate system is chosen that serves two purposes. It allows for accurate movement of model parameters throughout the grid space, and it helps simplify calculations. There are numerous types of coordinate systems that can be incorporated, from the familiar Cartesian, spherical, and cylindrical systems commonly in use, to the less obvious ones used in NWP like sigma (a type of terrain-following coordinate system), pressure, or temperature levels.

In the case of NAM model used here, the modelers chose a hybrid coordinate system where sigma and pressure level coordinates are used in conjunction to create a compromise between model performance in computation time and coordinate accuracy. With sigma coordinates, they improve calculation times at the lower boundary layers by gaining terrain following, while in the mid and upper layers where they use pressure levels, they improve coordinate accuracy.

As noted, models make estimates to gain advantages in calculation time without compromising accuracy. Still with a model implementing an array of errors into a chaotic environment that was not perfectly resolved to begin with, a long-range forecast cannot achieve perfection. So we will examine how close the estimate is when converted from the hybrid system into the World Geodetic System (WGS) coordinates, and figure out if the inaccuracies of those conversions really matter to the TDA, the last link of the knowledge extraction.

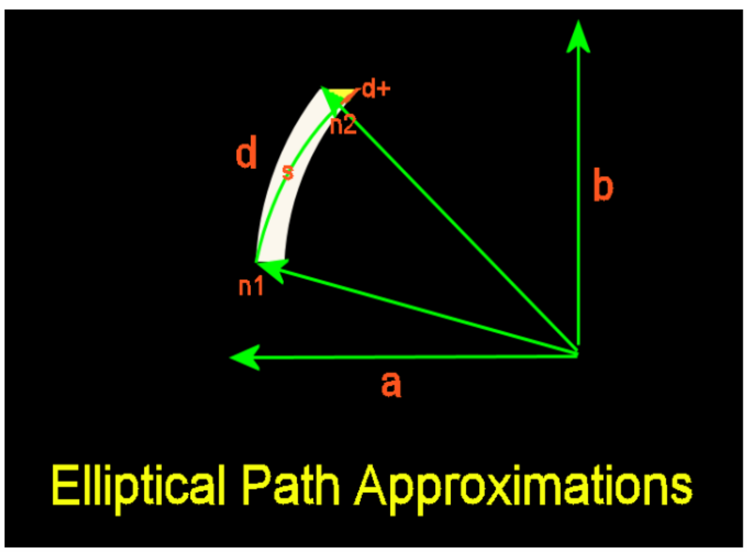
C. GRID EXTRACTIONS & RESOLUTION

The data for this project came from the NCEP's public FTP servers in WMO GRIB format. This is a packed data format that allows for large volumes of data to be transmitted using bundling techniques to reduce the data size. The downside is that it requires decoding and unpacking operations to get the data back to a usable grid. I chose to use MATLAB for this process since I found a decoder that worked for converting the files back to grids. For each individual resolution I needed to separate levels of winds available from the model data sets. I have included the list of grids extracted from each model resolution as additional information in the MATLAB code provided subsequently in this thesis.

Once the winds were extracted from the layer, a WGS84 consistent latitude and longitude grid needed to be calculated from the model information extracted from the GRIB file. This as it turned out was no simple task. The model information did provide data on grid resolution, number of grid points, and the geographic position of the lower left corner of the grid. It also noted that the data was projected in the Lambert Conformal standard. Research into Lambert Conformal showed that it was a conic projection, with the longitude at the center of the grid defining the grid spacing. It also defined the meridional grid spacing as constant. With that information at hand we had to devise a system to determine the longitude at each grid point along a meridian, and then find the center latitude value, to determine the Earth's polar radius at that point, for determining the coordinates of the longitudinal grid. Again this turned out to be no easy task. We wrote code to estimate the meridional latitudes based on grid space with reasonable accuracy and computational efficiency.

With coordinates to represent the grids, we could attach geographic information to the wind points so that bilinear interpolation could determine the wind values of a layer at any geographic point within the grid. It was now possible to build wind profiles. That was until we started working on the 12km grid data, where we discovered there was no elevation information to work from. In the previous data sets I was using geopotential height to represent the height above ground level for the winds. That data did not exist in the 12km data sets, but could be calculated, so we added a module to calculate heights when developing profiles.

With the profiles built we generated statistics against the real-world winds, calculating RMS errors for 1 meter interpolated increments of winds in both the x and y directions from the ground to 5km. The values seemed at first to be larger than expected, but after checking the model and the code against on select levels of the model by running single step calculations, we verified the profiler was generating acceptable data, and the statistic package was generating reasonable numbers.



$$\theta_1 \, given: N(\theta_1) = a / \sqrt{(1 - e^2 \sin^2(\theta_1))}$$
$$d \, given: \Delta\theta = d / N(\theta_1)$$
$$\theta_2 = \theta_1 + \Delta\theta$$
$$N(\theta_2) = a / \sqrt{(1 - e^2 \sin^2(\theta_2))}$$

$$\Delta N = N_1 - N_2; \;\; S = (d^2 - \Delta N^2)^{1/2}$$
$$S = r_a \Delta\theta; \;\; d / r_a = \theta_d$$
$$where \;\; e^2 = (a^2 - b^2) / a^2$$

1

Figure 7. Determination of Elliptical Path Length

D. LIMITING ASSUMPTIONS

Initially we were surprised by the error rates of comparing model forecast winds to measured winds recorded by the RAOBs, but after analyzing it more we began to realize that the results were not unreasonable. The initial thinking here was that the coordinates calculated from the model fields are a best estimate but that they may not directly match to the earth coordinates or the WGS84 coordinates used by the RAOB to record the actual wind measurements may have a large impact on the error rates. Our elliptic path integrals may not be as accurate as a GPS coordinate model, but I do believe they are close, at least to the same order of magnitude, as GPS.

Instead after some careful estimation we believe the best estimate for the difference between real-world values and the model values is that the model value is a snapshot of time. In the real world we have all seen winds change in a matter of minutes or hours. The real-world winds are a short time interval average, but nonetheless they are an average. Wind gust, or a lull in the wind can modify that average. What it really works out as, are that models smooth information in 4 dimensions, so it makes it difficult to compare them with real-world short-term average measurements. It is an unfair comparison for longer-term forecasts, but that was the purpose of this experiment to examine the extent and the effects of the long-range predictions.

VI. STATISTICS

A. THE RESEARCH

The purpose of this thesis was to examine the sensitivity of tactical decision aids to weather model data resolution with the assistance of software agents. Part of that examination was to inspect the system for errors, and determine if the magnitude of those errors degraded the results. Upon initial inspection of my statistics, I found a serious error in the analysis system. My statistics agent was designed to take the one-meter level increments of my profile winds and calculate the total root-mean square error throughout the column. When I examined those statistics I expected that as the model resolution increased there would be a decrease in error. This was not the case as my 32km model data sets had far greater errors than my 81km data sets. Even my 12km data was showing errors nearly as large as or slightly larger than those of the 81km data.

This indicated that there was no justification for high-resolution data to model tactical applications because there were no benefits to outputs. This would be highly counterintuitive. I rapidly began to suspect there was an error in the way I was mapping my model data onto global positions, maybe an error in my transformation to Lambert Conformal coordinates. It took me a couple of weeks to find my mistake was primarily conceptual. The GRIB standards for data lay them out in a grid from the southwest corner in a rectangular configuration. When they are laid on the spherical Earth they do not neatly wrap around the sphere with the grid points lying upon the parallels and meridians. Instead I had to remove a portion of my agent and replace it with new code that transformed rectangular coordinates of constant grid spacing using a Lambert Conformal conversion to geographic coordinates, thus associating accurately grid points to global locations. After double-checking my solutions I found that I had significantly changed my positioning error, and my final grid errors were now down to less than 0.1 degrees over a 120 degree range of longitudes.

Positioning error I discovered was significant well outside the GPS expectations of errors on the order of 1 to 3 meters. This I assumed was going to make a significant difference to my final accuracy. After slightly more analysis I realized that my

positioning of the release point within my grids could be off by several meters and I would still get a consistent profile from my agents. Even at the finest resolution there is more than 12,000 meters difference between data points. Interpolating between points at that resolution would not make more than a few ten thousandths of a meter per second difference in my profiles. Furthermore, that small of a difference would not matter given that the real world winds do not approximate model wind speeds anywhere near that accurately. Secondly, a thousandth of a meter per second speed difference integrated over 5000 levels of altitude should be less than a half of a meter difference in distances when the agent makes its calculations. To me that indicated that if I had target misses of less than 1.5 meters, I could expect that at the specified model resolution the tactical decision aid would be insensitive to difference between real world winds and model winds.

After re-running winds to generate profiles with my new coordinate system in place, I re-ran my root-mean square statistics agents. I chose to use estimates at each level of altitude and average them over the ranges occurring within the simulations to give me the best representation of the total error over the fall distance. During the descent phase of modeling the errors can be in both directions, with an overall effect of negating the wind differencing as the objects fall. By generating a root-mean square error through the profile, I can get an idea of the overall column errors, and a good representation of which resolution sets provides the best comparisons to real world winds.

As indicated by the statistics shown, the 12km winds have the lowest errors over the column as shown in Tables 1 through 6. One reason for this is the 12km data sets do not contain as many levels of data as the other columns, ending their information at around 1300 meters of altitude. As explained in Chapter I, winds within this range are typically slower, and represent lower potential to generate large errors. Additionally the errors showed no significant difference between 81km data and 32km data, which was to be expected when the resolution sets come from the same model.

The bomb and chute agent results, in Tables 7 through 9 and 10 through 12 respectively, confirmed what is heard most often about the models. While increased

resolution does give you a better picture of the overall flow pattern of the atmosphere, each individual model captures the state of the atmosphere rather well. The primary detriment to the forecast profiles was the progression of atmospheric changes through the data sets, which affected the directionality and intensity. If the model was late to bring the storm over the target area, then there were significant differences between the compared sets direction and speed. When there was a slow state of change in the true atmosphere, especially in the case of the bomb agent, all of the data sets were nearly good enough to show GPS precision for planning. Nevertheless, for the chute agent, real world winds are far too variable for long term accurate planning. Even measured winds, which are integrated into the model, undergo a large amount of temporal smoothing. The variability cannot be carried through the forecast, but that does not make the forecast less valuable.

Roanoke	12 km							Mean
	Forecasts							
Valid Time	Analysis	12	24	36	48	60	72	Mean
02140500	2.8321							2.83
02140512	3.0076	1.7183						2.36
02150500	3.2671	3.2615	3.6935					3.41
02150512	1.6365	1.3667	2.0380	2.5637				1.9
02160500	2.3049	1.9042	1.5187	1.1084	0.9500			1.56
02160512	2.2369	3.1029	5.2409	5.3153	7.5309	4.9109		4.72
02170500	3.0772	2.5287	2.6413	2.7576	2.4397	2.3470	2.2262	2.57
02170512	3.3883	2.9165	2.7196	2.5988	3.3394	1.9877	2.4640	2.77
02180500	0.7664	2.0990	2.8551	1.8724	0.7293	0.9144	1.8809	1.59
02180512	3.8794	1.5777	1.5288	1.7223	3.1412	3.5473	3.4641	2.69
04040512	3.3514	2.1026						2.73
04050500	1.3725	1.7033	1.0754					1.38
04050512	2.6211	2.7957	2.9098	2.3927				2.68
04060500	1.9368	1.6404	1.8043	1.2966	1.4015			1.62
04060512	2.2680	2.8344	4.3789	3.4150	4.9257	7.4591		4.21
04070500	1.7679	2.5781	1.6821	1.7978	1.9060	1.9394	2.8161	2.07
04070512	2.5566	3.3184	3.0341	2.9829	2.9171	3.1786	3.9155	3.13
04080500	1.3419	2.1690	4.0959	3.4994	3.5439	4.6467	6.0985	3.63
04080512	1.4432	1.2594	2.2078	3.0954	1.2202	1.9305	1.7254	1.84
Mean	2.37	2.27	2.71	2.6	2.84	3.29	3.07	

Table 1. RMS error between the Roanoke 12km resolution forecast and measured winds to 1300 meters above ground level.

Roanoke	32km							
	Forecasts							
Valid Time	Analysis	12	24	36	48	60	72	Mean
02140500	6.2585							6.26
02140512	8.3141	5.8062						7.06
02150500	7.6819	10.4799	11.8426					10
02150512	3.2966	3.1454	2.6453	1.9043				2.75
02160500	4.6163	4.0829	2.9199	3.2726	2.3755			3.45
02160512	4.7393	5.6911	6.5574	6.7377	8.7829	7.3961		6.65
02170500	6.4265	6.4136	5.5970	5.0073	7.6097	7.7666	6.9527	6.54
02170512	6.8501	7.2000	6.2059	5.6299	6.5530	6.6068	6.0232	6.44
02180500	5.8909	5.7084	5.9977	6.0180	7.2834	7.5010	9.1043	6.79
02180512	8.0454	7.2322	7.1610	5.5816	7.0463	7.3644	8.8505	7.33
04040512	7.0355	6.1957						6.62
04050500	4.0411	4.7994	5.1409					4.66
04050512	2.4761	2.6319	3.0972	3.0886				2.82
04060500	3.1683	3.7565	3.4966	2.7665	2.9011			3.22
04060512	2.6855	3.2786	3.9672	3.6654	5.0312	6.3547		4.16
04070500	4.4459	4.3123	4.2275	3.5394	4.2890	3.4614	5.5409	4.26
04070512	5.7171	5.5490	6.5740	5.0418	4.9732	7.3654	6.8494	6.01
04080500	5.3164	6.3650	8.7900	5.0798	4.7777	6.2107	8.4914	6.43
04080512	1.9930	1.7774	2.7630	3.7238	2.3169	2.7437	2.7655	2.58
Mean	5.21	5.25	5.44	4.36	5.33	6.28	6.82	

Table 2. RMS error between the Roanoke 32km resolution forecast and measured winds to 4800 meters above ground level.

Roanoke	81km Forecasts							
Valid Time	Analysis	12	24	36	48	60	72	Mean
02140500	4.4987							4.5
02140512	3.3374	5.2426						4.29
02150500	5.4930	8.2035	8.9914					7.56
02150512	0.9946	1.2145	1.8230	1.9512				1.5
02160500	2.6195	2.5300	2.0600	1.7861	3.1025			2.42
02160512	2.9605	3.7110	4.9914	4.2927	6.3024	6.2415		4.75
02170500	4.7633	4.5001	3.9918	3.5463	4.9043	5.0540	5.5253	4.61
02170512	4.1241	3.9788	3.2960	3.1937	3.8164	2.9407	2.7866	3.45
02180500	2.7787	2.5858	2.6251	2.7967	4.0805	4.2282	5.4151	3.5
02180512	6.0760	5.6406	6.1106	6.1210	3.9272	4.5047	5.2163	5.37
04040512	4.3069	3.4638						3.89
04050500	1.6280	2.1091	2.2684					2
04050512	3.4178	2.4000	2.5732	3.2076				2.9
04060500	1.6869	2.1033	2.1983	1.9750	2.0971			2.01
04060512	2.1697	2.6834	3.7142	2.6537	3.8832	6.4378		3.59
04070500	2.5177	2.2500	2.0689	1.8655	2.0027	2.2322	4.9611	2.56
04070512	3.9243	4.8108	3.8699	3.3893	3.3668	4.7094	5.8161	4.27
04080500	3.6679	4.3495	5.9705	4.1249	4.0062	5.2967	7.0442	4.92
04080512	2.3314	2.3430	2.5291	3.0677	2.1491	2.7003	2.0937	2.46
Mean	3.33	3.56	3.69	3.14	3.64	4.43	4.86	

Table 3. RMS error between the Roanoke 81km resolution forecast and measured winds to 4800 meters above ground level.

Vandenberg	12km							
	Forecasts							
Valid Time	Analysis	12	24	36	48	60	72	Mean
02140500	2.9644							2.96
02140512	1.3261	2.0498						1.69
02150500	1.6851	4.3700	2.5324					2.86
02150512	2.0225	3.5804	3.7287	3.4898				3.21
02160500	3.0284	5.7159	4.9767	3.9935	5.3025			4.6
02160512	3.9035	6.0498	7.5015	7.4269	7.8370	8.2498		6.83
02170500	2.1554	3.2939	4.2418	5.6336	4.6435	5.4246	4.8896	4.33
02170512	2.5483	4.6809	5.1224	5.2106	5.2110	4.3551	5.4903	4.66
02180500	3.6833	7.2414	7.2493	6.3518	5.9157	7.1305	6.8823	6.35
02180512	1.7493	3.6020	6.0550	4.1666	3.9336	2.3546	6.1091	4
04040500	2.2554							2.26
04040512	2.3665	3.9079						3.14
04050500	5.9456	6.2423	6.4521					6.21
04050512	5.5039	2.8195	3.5494	3.8354				3.93
04060500	1.7424	1.8920	1.5907	2.9110	1.6732			1.96
04060512	2.8509	2.6047	2.7278	2.4010	4.1440	2.2559		2.83
04070500	0.9718	2.9686	2.7766	2.3408	3.1460	5.3112	2.7836	2.9
04070512	2.9788	3.4642	4.4564	4.7275	6.5111	6.0314	6.8920	5.01
04080500	2.6120	2.7962	2.7921	4.1781	2.8717	2.7753	3.3644	3.06
04080512	4.2557	4.7533	5.1854	4.2819	5.5410	5.2806	5.8918	5.03
Mean	2.83	4	4.43	4.35	4.73	4.92	5.29	

Table 4. RMS error between the Vandenberg 12km resolution forecast and measured winds to 1300 meters above ground level.

Vandenberg	32km							
	Forecasts							
Valid Time	Analysis	12	24	36	48	60	72	Mean
02140500	2.8585							2.86
02140512	2.1447	3.8732						3.01
02150500	1.8182	3.3544	1.9602					2.38
02150512	2.3351	4.5378	3.7723	4.3545				3.75
02160500	2.6077	5.4782	3.7435	2.9292	3.1695			3.59
02160512	4.3780	5.1037	6.4432	6.0341	6.0100	5.8367		5.63
02170500	2.5134	3.2223	4.9578	4.5919	4.5973	5.9234	5.0020	4.4
02170512	3.0738	3.8740	4.1693	4.3316	4.8457	3.4967	5.0723	4.12
02180500	4.4009	9.3619	7.6547	7.6269	7.4915	8.5401	8.4251	7.64
02180512	2.8315	3.4342	4.9241	3.4656	3.8475	3.9038	4.9620	3.91
04040500	2.7408							2.74
04040512	3.8402	5.7918						4.82
04050500	6.1761	6.3854	6.7315					6.43
04050512	3.4340	2.2346	2.7608	2.6705				2.77
04060500	2.8783	3.1688	3.1941	3.7525	3.1549			3.23
04060512	2.0596	2.9204	2.5998	3.2085	4.8803	3.4554		3.19
04070500	2.1940	3.8609	3.6466	3.1806	4.2103	4.8218	3.0916	3.57
04070512	3.6430	4.1995	6.2812	4.0439	6.1749	5.6660	6.2808	5.18
04080500	4.2072	4.5600	4.3184	5.8384	3.7096	4.4303	8.2685	5.05
04080512	2.8951	4.3988	4.7600	4.5113	5.3463	5.0538	5.6605	4.66
Mean	3.32	4.43	4.49	4.32	4.79	5.11	5.85	

Table 5. RMS error between the Vandenberg 32km resolution forecast and measured winds to 4800 meters above ground level.

Vandenberg	81km							
	Forecasts							
Valid Time	Analysis	12	24	36	48	60	72	Mean
02140500	3.4282							3.43
02140512	1.9694	3.6329						2.8
02150500	1.9652	3.1350	2.1605					2.42
02150512	2.5989	4.3518	3.4128	4.0887				3.61
02160500	2.2156	4.6069	4.3298	3.3077	3.6303			3.62
02160512	4.4933	5.8833	6.7996	6.3001	6.2352	5.8972		5.93
02170500	2.8937	3.5971	4.6664	4.9081	4.5582	5.0246	4.8836	4.36
02170512	2.8031	3.5135	4.3700	4.0745	4.6266	3.9021	5.2122	4.07
02180500	5.0377	9.5535	7.6624	8.2093	8.6091	8.5364	7.9664	7.94
02180512	2.5912	3.9008	5.5926	4.3884	4.8474	3.8755	4.9854	4.31
04040500	1.9742							1.97
04040512	4.5601	5.6194						5.09
04050500	6.3262	6.2059	6.6245					6.39
04050512	3.5501	2.2842	3.0418	3.2798				3.04
04060500	2.0388	2.7262	3.3296	3.4698	2.5166			2.82
04060512	2.7236	2.6047	2.6594	3.1941	4.4200	3.5262		3.19
04070500	1.9799	3.5512	3.4276	3.1743	4.0899	4.5613	2.9929	3.4
04070512	2.9484	3.4724	4.9610	2.6787	5.2299	4.6951	4.8435	4.12
04080500	4.0041	4.3834	4.2071	5.9800	3.7305	4.8101	9.2308	5.19
04080512	3.5438	5.1050	5.5764	5.2078	6.0007	5.7458	5.8985	5.3
Mean	3.35	4.34	4.55	4.45	4.87	5.06	5.75	

Table 6. RMS error between the Vandenberg 81km resolution forecast and measured winds to 4800 meters above ground level.

Bomb Agent Output	Forecasts															
12km 1300m drop Roanoke	Rawin		Analysis		12 hour		24 hour		36 hour		48 hour		60 hour		72 hour	
Valid Time	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
02140500	0.42	1113.7	-0.1	1113.5												
Delta			-0.52	-0.2												
02140512	0.48	1114.1	0.06	1113.5	0.61	1114.4										
Delta			-0.42	-0.6	0.13	0.3										
02150500	2.35	1112.6	1.93	1113	2.07	1112.8	2.1	1112.8								
Delta			-0.42	0.4	-0.28	0.2	-0.25	0.2								
02150512	1.32	1111.2	1.23	1111.4	1.27	1110.9	1.42	1111	1.65	1110.7						
Delta			-0.09	0.2	-0.05	-0.3	0.1	-0.2	0.33	-0.5						
02160500	0.78	1112.6	0.33	1113.2	0.42	1112.9	0.72	1113.2	0.72	1113.1	0.87	1113.1				
Delta			-0.45	0.6	-0.36	0.3	-0.06	0.6	-0.06	0.5	0.09	0.5				
02160512	2.63	1111.4	2.3	1111.9	2.49	1111.9	2.32	1112.8	2.05	1112.3	1.92	1113.3	2.39	1112.4		
Delta			-0.33	0.5	-0.14	0.5	-0.31	1.4	-0.58	0.9	-0.71	1.9	-0.24	1		
02170500	2.03	1110.5	1.35	1110.9	1.44	1110.5	1.63	1110.3	1.58	1110	1.58	1111	1.8	1111.1	1.65	1110.6
Delta			-0.68	0.4	-0.59	0	-0.4	-0.2	-0.45	-0.5	-0.45	0.5	-0.23	0.6	-0.38	0.1
02170512	1.67	1110.7	1.29	1110.9	1.42	1111	1.53	1110.9	1.53	1110.7	1.16	1111.1	1.84	1110.8	1.77	1110.5
Delta			-0.38	0.2	-0.25	0.3	-0.14	0.2	-0.14	0	-0.51	0.4	0.17	0.1	0.1	-0.2
02180500	1.93	1111.4	1.82	1111.4	2.27	1110.6	2.26	1110.4	2.06	1110.7	1.9	1111.1	2.05	1111.3	2.33	1111
Delta			-0.11	0	0.34	-0.8	0.33	-1	0.13	-0.7	-0.03	-0.3	0.12	-0.1	0.4	-0.4
02180512	2.58	1109.6	2.22	1110.3	2.58	1109.5	2.76	1109.6	2.62	1109.4	2.19	1109.6	2.52	1110.1	2.38	1110.4
Delta			-0.36	0.7	0	-0.1	0.18	0	0.04	-0.2	-0.39	0	-0.06	0.5	-0.2	0.8
Valid Time																
04040512	2.73	1109.4	2.38	1110.2												
Delta			-0.35	0.8												
04050500	1.36	1111.1	1.45	1110.9	1.59	1111.1										
Delta			0.09	-0.2	0.23	0										
04050512	0.6	1111	0.59	1111.6	0.52	1111.7	0.7	1111.9								
Delta			-0.01	0.6	-0.08	0.7	0.1	0.9								
04060500	0.35	1113.2	0.19	1113.3	0.29	1113.6	0.26	1113.3	0.38	1113.5						
Delta			-0.16	0.1	-0.06	0.4	-0.09	0.1	0.03	0.3						
04060512	0.75	1111.7	1.08	1112.1	1.33	1111.8	1.52	1112.5	1.08	1112.3	1.32	1112.3				
Delta			0.33	0.4	0.58	0.1	0.77	0.8	0.33	0.6	0.57	0.6				
04070500	0.38	1113.2	0.35	1113	0.85	1113.3	0.51	1113.4	0.51	1112.7	0.35	1113.6	0.5	1113		
Delta			-0.03	-0.2	0.47	0.1	0.13	0.2	0.13	-0.5	-0.03	0.4	0.12	-0.2		
04070512	1.14	1113.9	0.76	1113.7	0.94	1113.4	0.61	1114.1	0.8	1113.5	0.74	1113.6	0.41	1114.1	0.65	1113.4
Delta			-0.38	-0.2	-0.2	-0.5	-0.53	0.2	-0.34	-0.4	-0.4	-0.3	-0.73	0.2	-0.49	-0.5
04080500	0.23	1112.7	0.34	1112.7	0.52	1112.7	0.56	1113.3	0.66	1113.3	1.01	1112.5	0.89	1113.5	0.91	1113.8
Delta			0.11	0	0.29	0	0.33	0.6	0.43	0.6	0.78	-0.2	0.66	0.8	0.68	1.1
04080512	-0.47	1111.5	-0.34	1111.3	-0.55	1111.4	-0.08	1111.5	0.14	1111.8	-0.32	1111.8	-0.51	1111.4	-0.4	1111.4
Delta			0.13	-0.2	-0.08	-0.1	0.39	0	0.61	0.3	0.15	0.3	-0.04	-0.1	0.07	-0.1

Table 7. Comparison of bomb agent outputs using Roanoke 12km forecast and measured winds to 1300 meters above ground level.

Bomb Agent Output 32km 4800m drop Roanoke

Forecasts

	Rawin		Analysis		12 hour		24 hour		36 hour		48 hour		60 hour		72 hour	
Valid Time	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
02140500	19.67	3187.3	14.33	3196.4												
Delta			-5.34	9.1												
02140512	18.58	3203.1	10.4	3211.5	10.96	3206.8										
Delta			-8.18	8.4	-7.62	3.7										
02150500	31.74	3181.1	25.28	3195	19.03	3196.9	16.86	3197.8								
Delta			-6.46	13.9	-12.71	15.8	-14.88	16.7								
02150512	15.51	3181.5	12.84	3186.9	13.55	3185.7	14.71	3184.8	14.72	3183.7						
Delta			-2.67	5.4	-1.96	4.2	-0.8	3.3	-0.79	2.2						
02160500	14.37	3194.1	9.43	3199.5	8.32	3196.4	10.77	3196.1	10.7	3197.2	13.34	3192.5				
Delta			-4.94	5.4	-6.05	2.3	-3.6	2	-3.67	3.1	-1.03	-1.6				
02160512	19.37	3188.3	15.3	3193.2	15.47	3196.5	16.57	3196.2	14.3	3194.6	13.17	3199	15.28	3200		
Delta			-4.07	4.9	-3.9	8.2	-2.8	7.9	-5.07	6.3	-6.2	10.7	-4.09	11.7		
02170500	28.56	3179.2	23.97	3189.6	24.08	3190.4	24.35	3188.8	24.95	3187.5	19.31	3189.4	21.39	3190.8	24.77	3192
Delta			-4.59	10.4	-4.48	11.2	-4.21	9.6	-3.61	8.3	-9.25	10.2	-7.17	11.6	-3.79	12.8
02170512	28.43	3182.1	21.87	3194	20.93	3193.8	24.01	3193.8	24.52	3193	21.64	3192.7	22.41	3194	23.14	3192.8
Delta			-6.56	11.9	-7.5	11.7	-4.42	11.7	-3.91	10.9	-6.79	10.6	-6.02	11.9	-5.29	10.7
02180500	32.02	3163.3	32.1	3177.3	33.51	3176.7	32.68	3178.4	32.77	3178.1	33.28	3180.2	30.33	3183.7	35.38	3184.6
Delta			0.08	14	1.49	13.4	0.66	15.1	0.75	14.8	1.26	16.9	-1.69	20.4	3.36	21.3
02180512	39.84	3167.3	33.93	3183.3	33.01	3179.8	32.4	3177.7	33.17	3173.1	34.87	3181.1	36.03	3181.8	31.58	3182.6
Delta			-5.91	16	-6.83	12.5	-7.44	10.4	-6.67	5.8	-4.97	13.8	-3.81	14.5	-8.26	15.3
Valid Time																
04040512	22.64	3162.4	25.86	3171.6												
Delta			3.22	9.2												
04050500	15.03	3167.4	16.93	3175.1	19.32	3175.6										
Delta			1.9	7.7	4.29	8.2										
04050512	6.7	3180.7	7.98	3179.9	7.56	3183.5	8.08	3183.9								
Delta			1.28	-0.8	0.86	2.8	1.38	3.2								
04060500	9.74	3188.1	7.12	3190	6.75	3192.3	7.6	3191.7	8.08	3192.1						
Delta			-2.62	1.9	-2.99	4.2	-2.14	3.6	-1.66	4						
04060512	6.49	3187.2	5.38	3189.4	7.11	3190.9	8.12	3190.7	6.06	3192.9	6.96	3193.9				
Delta			-1.11	2.2	0.62	3.7	1.63	3.5	-0.43	5.7	0.47	6.7				
04070500	9.99	3194.5	4.22	3197.9	5.41	3196.9	4.89	3199	4.8	3196.4	5.31	3198.6	7.13	3194.4		
Delta			-5.77	3.4	-4.58	2.4	-5.1	4.5	-5.19	1.9	-4.68	4.1	-2.86	-0.1		
04070512	11.71	3205.5	1.97	3208.1	4.5	3203.7	0.17	3208.1	3.11	3204.3	3.09	3203.9	-0.41	3209.7	-0.58	3203.4
Delta			-9.74	2.6	-7.21	-1.8	-11.54	2.6	-8.6	-1.2	-8.62	-1.6	-12.12	4.2	-12.29	-2.1
04080500	7.25	3187.6	3.57	3196.3	5.18	3200.8	1.19	3204.2	4.5	3196.5	5.34	3193.9	5.52	3197.8	4.97	3204.1
Delta			-3.68	8.7	-2.07	13.2	-6.06	16.6	-2.75	8.9	-1.91	6.3	-1.73	10.2	-2.28	16.5
04080512	-3.53	3185	-1.78	3183.5	-2.64	3184.5	-1.18	3183.2	-0.41	3189	-0.66	3185.5	-1.19	3187.1	-0.75	3185.7
Delta			1.75	-1.5	0.89	-0.5	2.35	-1.8	3.12	4	2.87	0.5	2.34	2.1	2.78	0.7

Table 8. Comparison of bomb agent outputs using Roanoke 32km forecast and measured winds to 4800 meters above ground level.

Bomb Agent Output 81km 4800m drop Roanoke

Forecasts	Rawin		Analysis		12 hour		24 hour		36 hour		48 hour		60 hour		72 hour	
Valid Time	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
02140500	19.67	3187.3	15.42	3190.9												
Delta			-4.25	3.6												
02140512	18.58	3203.1	14.46	3204.5	11.6	3201.7										
Delta			-4.12	1.4	-6.98	-1.4										
02150500	31.74	3181.4	26.72	3190.3	22.09	3194.3	20.36	3194.1								
Delta			-5.02	8.9	-9.65	12.9	-11.38	12.7								
02150512	15.51	3181.5	14.48	3182	14.89	3180.6	17.59	3179.9	17.25	3178.7						
Delta			-1.03	0.5	-0.62	-0.9	2.08	-1.6	1.74	-2.8						
02160500	14.37	3194.1	13.04	3197.2	12.04	3194.1	14.48	3193.73	14.6	3194.5	17.1	3189.6				
Delta			-1.33	3.1	-2.33	0	0.11	-0.37	0.23	0.4	2.73	-4.5				
02160512	19.37	3188.3	18.66	3191.4	19.78	3193.8	21.44	3193.5	19.19	3191.6	18.72	3198.1	23	3197.5		
Delta			-0.71	3.1	0.41	5.5	2.07	5.2	-0.18	3.3	-0.65	9.8	3.63	9.2		
02170500	28.56	3179.2	25.83	3183.7	26.89	3183.3	28.93	3181	27.45	3179.4	24.31	3185	25.74	3184.9	26.85	3185.3
Delta			-2.73	4.5	-1.67	4.1	0.37	1.8	-1.11	0.2	-4.25	5.8	-2.82	5.7	-1.71	6.1
02170512	28.43	3182.1	26.37	3189.4	25.45	3188.8	27.83	3188.1	28.08	3186.4	26.03	3187.4	28.24	3187.5	28.53	3186.3
Delta			-2.06	7.3	-2.98	6.7	-0.6	6	-0.35	4.3	-2.4	5.3	-0.19	5.4	0.1	4.2
02180500	32.02	3163.3	32.7	3166.9	33.32	3166.1	32.69	3168	33.94	3167.4	35.48	3169.5	32.8	3173	34.53	3174.8
Delta			0.68	3.6	1.3	2.8	0.67	4.7	1.92	4.1	3.46	6.2	0.78	9.7	2.51	11.5
02180512	39.84	3167.3	32.11	3173.4	30.79	3170.1	30	3167.4	31.15	3163.2	35.96	3170.2	37.03	3170.6	33.89	3172.4
Delta			-7.73	6.1	-9.05	2.8	-9.84	0.1	-8.69	-4.1	-3.88	2.9	-2.81	3.3	-5.95	5.1
Valid Time																
04040512	22.64	3162.4	24.53	3162.8												
Delta			1.89	0.4												
04050500	15.03	3167.4	16.38	3167.9	18.56	3167.2										
Delta			1.35	0.5	3.53	-0.2										
04050512	6.7	3180.7	9.05	3176.1	8.41	3180.3	8.53	3180.2								
Delta			2.35	-4.6	1.71	-0.4	1.83	-0.5								
04060500	9.74	3188.1	9.91	3188.2	10.92	3191.2	11.06	3189.6	11.66	3189.5						
Delta			0.17	0.1	1.18	3.1	1.32	1.5	1.92	1.4						
04060512	6.49	3187.2	8.96	3187.6	10.62	3187.8	12.46	3188	8.67	3190.6	9.99	3190.9				
Delta			2.47	0.4	4.13	0.6	5.97	0.8	2.18	3.4	3.5	3.7				
04070500	9.99	3194.5	7.33	3196.4	8.97	3194.4	8.58	3196	8.22	3193.3	9.46	3195.6	10.23	3190.6		
Delta			-2.66	1.9	-1.02	-0.1	-1.41	1.5	-1.77	-1.2	-0.53	1.1	0.24	-3.9		
04070512	11.71	3205.5	7.6	3207.7	10.48	3203.4	5.65	3205.9	7.62	3203.7	8.21	3202.8	5.22	3208.8	3.4	3202.4
Delta			-4.11	2.2	-1.23	-2.1	-6.06	0.4	-4.09	-1.8	-3.5	-2.7	-6.49	3.3	-8.31	-3.1
04080500	7.25	3187.6	4.67	3193.6	7.96	3194.6	6.08	3198.6	7.3	3191.7	8.56	3191.3	9.21	3193	8.91	3200.8
Delta			-2.58	6	0.71	7	-1.17	11	0.05	4.1	1.31	3.7	1.96	5.4	1.66	13.2
04080512	-3.53	3185	-2.02	3180.2	-3.52	3181.5	-1.19	3185	0.68	3184.6	-1.1	3184.6	0.34	3185.2	-0.96	3184.9
Delta			1.51	-4.8	0.01	-3.5	2.34	0	4.21	-0.4	2.43	-0.4	3.87	0.2	2.57	-0.1

Table 9. Comparison of bomb agent outputs using Roanoke 81km forecast and measured winds to 4800 meters above ground level.

Chute Agent Output Forecasts 12km 300m drop Vandenburg

Valid Time	Rawin X	Rawin Y	Analysis X	Analysis Y	12 hour X	12 hour Y	24 hour X	24 hour Y	36 hour X	36 hour Y	48 hour X	48 hour Y	60 hour X	60 hour Y	72 hour X	72 hour Y
02140500	163.9	328.3	9.8	259.4												
Delta			-154.1	-68.9												
02140512	2.9	208.4	0.9	150.2	-9.3	139.2										
Delta			-2	-58.2	-12.2	-69.2										
02150500	6.6	253.7	35.3	249.9	90	113.1	68.4	199.2								
Delta			28.7	-3.8	83.4	-140.6	61.8	-54.5								
02150512	-174.2	330.8	-51.2	287.8	-60	212.6	-52.5	183.7	-44	218.5						
Delta			123	-43	114.2	-118.2	121.7	-147.1	130.2	-112.3						
02160500	-285.2	520.2	-63.1	450.8	-59.7	303.5	-53.8	317.3	-50	398.6	-6.9	359				
Delta			222.1	-69.4	225.5	-216.7	231.4	-202.9	235.2	-121.6	278.3	-161.2				
02160512	-429.7	475.6	-121.7	385	-62.2	321.6	-26.1	280.1	-35.3	253.9	-47.6	233.8	-2	245.8		
Delta			308	-90.6	367.5	-154	403.6	-195.5	394.4	-221.7	382.1	-241.8	427.7	-229.8		
02170500	-202.5	471.3	-63.8	359.8	-47.7	288	-14.2	301	21.1	230.1	10.1	271.7	-6.1	243.2	24.8	281.8
Delta			138.7	-111.5	154.8	-183.3	188.3	-170.3	223.6	-241.2	212.6	-199.6	196.4	-228.1	227.3	-189.5
02170512	-64.3	43.3	-116.2	171.7	-46.6	256.3	-37	227.7	-83.5	265.3	-71.8	245	-63.5	197.6	-48.1	220.6
Delta			-51.9	128.4	17.7	213	27.3	184.4	-19.2	222	-7.5	201.7	0.8	154.3	16.2	177.3
02180500	-278.8	150.5	-115	243.4	13.1	72.8	24.5	253.5	0	276.4	-58	388.9	-5.6	356.8	-48.7	271.9
Delta			163.8	92.9	291.9	-77.7	303.3	103	278.8	125.9	220.8	238.4	273.2	206.3	230.1	121.4
02180512	-220.3	514.7	-141.1	419.6	-85	312.6	5.25	281.5	-77.4	300.8	-93.9	286.7	-137.8	361.9	-26.9	232.4
Delta			79.2	-95.1	135.3	-202.1	225.55	-233.2	142.9	-213.9	126.4	-228	82.5	-152.8	193.4	-282.3
Valid Time																
04040500	176.4	90.5	186.2	-36.3												
Delta			9.8	-126.8												
04040512	116.5	-100.3	101	-94.5	23.7	-32.4										
Delta			-15.5	5.8	-92.8	67.9										
04050500	269.4	-172	146.6	-93.9	160.8	-152.7	122.2	-106.8								
Delta			-122.8	78.1	-108.6	19.3	-147.2	65.2								
04050512	-73.3	-45.3	-200.5	57.7	-42.4	-61.8	-49.8	-50	-81.5	-1.7						
Delta			-127.2	103	30.9	-16.5	23.5	-4.7	-8.2	43.6						
04060500	165.4	103.5	75	129	48.1	102.2	100.5	112.8	104	24.7	103.2	78.5				
Delta			-90.4	25.5	-117.3	-1.3	-64.9	9.3	-61.4	-78.8	-62.2	-25				
04060512	55.5	104.1	-49.2	156.7	-72.3	140	-44.5	170.7	-74.6	134.4	-38.6	2.6	-25.7	95.5		
Delta			-104.7	52.6	-127.8	35.9	-100	66.6	-130.1	30.3	-94.1	-101.5	-81.2	-8.6		
04070500	140	92.3	145.8	129.4	120.5	-10.4	58.9	29.5	137.7	50	133.8	-10.8	135.5	-69.2	141.7	-0.5
Delta			5.8	37.1	-19.5	-102.7	-81.1	-62.8	-2.3	-42.3	-6.2	-103.1	-4.5	-161.5	1.7	-92.8
04070512	-65.4	99.7	24	85.8	32	32.8	19	-36.6	27	-47.8	41	-82	47.2	-85.8	19.4	-88.2
Delta			89.4	-13.9	97.4	-66.9	84.4	-136.3	92.4	-147.5	106.4	-181.7	112.6	-185.5	84.8	-187.9
04080500	293.1	-219.4	204.1	-182	230.3	-186.2	210.1	-166	229.5	-223.8	224.1	-214	241	-208.1	216.5	-130
Delta			-89	37.4	-62.8	33.2	-83	53.4	-63.6	-4.4	-69	5.4	-52.1	11.3	-76.6	89.4
04080512	51.9	-16.3	29.9	-141.4	27.9	-154.7	27.2	-156.1	64.4	-144.5	18	-182.7	37.4	-190.1	48.9	-218.5
Delta			-22	-125.1	-24	-138.4	-24.7	-139.8	12.5	-128.2	-33.9	-166.4	-14.5	-173.8	-3	-202.2

Table 10. Comparison of chute agent outputs using Vandenberg 12km forecast and measured winds to 300 meters above ground level.

Chute Agent Output 32km 300m drop Vandenburg

Valid Time	Rawin X	Rawin Y	Analysis X	Analysis Y	12 hour X	12 hour Y	24 hour X	24 hour Y	36 hour X	36 hour Y	48 hour X	48 hour Y	60 hour X	60 hour Y	72 hour X	72 hour Y
			Forecasts													
02140500	163.9	328.3	-52.6	279												
Delta			-216.5	-49.3												
02140512	2.9	208.4	26.1	152.3	8.6	133.3										
Delta			23.2	-56.1	5.7	-75.1										
02150500	6.6	253.7	-21.6	318.4	82	149.8	21.2	230.8								
Delta			-28.2	64.7	75.4	-103.9	14.6	-22.9								
02150512	-174.2	330.8	-94.6	362.9	-67.4	258.9	-51.7	223.4	-53.9	266.8						
Delta			79.6	32.1	106.8	-71.9	122.5	-107.4	120.3	-64						
02160500	-285.2	520.2	-182.5	550.5	-153.7	382.8	-139.7	392.9	-159.4	456.6	-87.9	400.5				
Delta			102.7	30.3	131.5	-137.4	145.5	-127.3	125.8	-63.6	197.3	-119.7				
02160512	-429.7	475.6	-224.1	465.7	-156	376.8	-84.6	330.2	-65.3	292.7	-99	267.9	-42.3	282.7		
Delta			205.6	-9.9	273.7	-98.8	345.1	-145.4	364.4	-182.9	330.7	-207.7	387.4	-192.9		
02170500	-202.5	471.3	-166.7	417.1	-144.7	340.4	-118.6	353.2	-45.8	275.7	-56.6	308.8	-82.6	277.3	-60.4	309
Delta			35.8	-54.2	57.8	-130.9	83.9	-118.1	156.7	-195.6	145.9	-162.5	119.9	-194	142.1	-162.3
02170512	-64.3	43.3	-114.5	179.5	-86.4	300.4	-72.9	249.2	-149.7	306.9	-125.4	303.8	-65.6	234.2	-79.8	283.6
Delta			-50.2	136.2	-22.1	257.1	-8.6	205.9	-85.4	263.6	-61.1	260.5	-1.3	190.9	-15.5	240.3
02180500	-278.8	150.5	-170.1	293.6	20	73.6	-8.1	282.2	-65.2	304.3	-149.3	408.5	-96.9	413.9	-88	311.4
Delta			108.7	143.1	298.8	-76.9	270.7	131.7	213.6	153.8	129.5	258	181.9	263.4	190.8	160.9
02180512	-220.3	514.7	-233.7	477.7	-167.5	358.1	-51.4	329	-149	354.6	-187.7	359.5	-215.7	454.9	-61	268.8
Delta			-13.4	-37	52.8	-156.6	168.9	-185.7	71.3	-160.1	32.6	-155.2	4.6	-59.8	159.3	-245.9
Valid Time																
04040500	176.4	90.5	206.6	-101.7												
Delta			30.2	-192.2												
04040512	116.5	-100.3	142.8	-178.6	60.5	-115.1										
Delta			26.3	-78.3	-56	-14.8										
04050500	269.4	-172	160.4	-173.7	180.3	-251.3	169.9	-220.9								
Delta			-109	-1.7	-89.1	-79.3	-99.5	-48.9								
04050512	-73.3	-45.3	-204.5	-32.9	-31.3	-179.2	-37.8	-162.5	-85.4	-102.4						
Delta			-131.2	12.4	42	-133.9	35.5	-117.2	-12.1	-57.1						
04060500	165.4	103.5	46.9	132.2	32.1	72.1	75.4	67.1	79.1	-63.1	80.1	6.4				
Delta			-118.5	28.7	-133.3	-31.4	-90	-36.4	-86.3	-166.6	-85.3	-97.1				
04060512	55.5	104.1	-40.3	99.7	-60.6	74	-33.1	140	-68.6	84.2	-17.9	-130.4	-10.1	14.9		
Delta			-95.8	-4.4	-116.1	-30.1	-88.6	35.9	-124.1	-19.9	-73.4	-234.5	-65.6	-89.2		
04070500	140	92.3	143.7	34.6	110	-138.1	46.3	-70	122.4	-68.3	125.2	-137.5	131.1	-202.1	138.5	-116.4
Delta			3.7	-57.7	-30	-230.4	-93.7	-162.3	-17.6	-160.6	-14.8	-229.8	-8.9	-294.4	-1.5	-208.7
04070512	-65.4	99.7	66.9	19.7	90.8	-27.9	73.5	-125.2	94.9	-138.6	70.6	-193.8	81.9	-194.6	53.4	-190.6
Delta			132.3	-80	156.2	-127.6	138.9	-224.9	160.3	-238.3	136	-293.5	147.3	-294.3	118.8	-290.3
04080500	293.1	-219.4	203.3	-280.7	235.4	-289.3	214.5	-254.9	249.4	-349.2	232.1	-328.1	264.8	-342	221.9	-263
Delta			-89.8	-61.3	-57.7	-69.9	-78.6	-35.5	-43.7	-129.8	-61	-108.7	-28.3	-122.6	-71.2	-43.6
04080512	51.9	-16.3	78.4	-220.5	80.4	-242.6	57.4	-240.5	105	-241	57.3	-295.1	79.6	-295.9	90.1	-337.4
Delta			26.5	-204.2	28.5	-226.3	5.5	-224.2	53.1	-224.7	5.4	-278.8	27.7	-279.6	38.2	-321.1

Table 11. Comparison of chute agent outputs using Vandenberg 32km forecast and measured winds to 300 meters above ground level.

Chute Agent Output 81km 300m drop Vandenburg
Forecasts

	Rawin		Analysis		12 hour		24 hour		36 hour		48 hour		60 hour		72 hour	
Valid Time	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
02140500	163.9	328.3	0.7	260.4												
Delta			-163.2	-67.9												
02140512	2.9	208.4	-7	185.9	-17.2	162.4										
Delta			-9.9	-22.5	-20.1	-46										
02150500	6.6	253.7	-53.6	282.6	4.4	170.6	22.2	200.4								
Delta			-60.2	28.9	-2.2	-83.1	15.6	-53.3								
02150512	-174.2	330.8	-54.8	318.2	-82.4	238.3	-67.1	220.3	-45.4	235.9						
Delta			119.4	-12.6	91.8	-92.5	107.1	-110.5	128.8	-94.9						
02160500	-285.2	520.2	-104.8	458.4	-87.5	321.8	-62.7	303.6	-26.9	359.1	-9.9	323.8				
Delta			180.4	-61.8	197.7	-198.4	222.5	-216.6	258.3	-161.1	275.3	-196.4				
02160512	-429.7	475.6	-98.6	390.5	-38.8	306.1	-13.9	285.8	-13	253	-25.5	259	3.7	272.6		
Delta			331.1	-85.1	390.9	-169.5	415.8	-189.8	416.7	-222.6	404.2	-216.6	433.4	-203		
02170500	-202.5	471.3	-40.2	338.9	-58.7	261.5	-25.2	263.3	-17	216.3	-12.4	249.9	-45	240	18.6	256.3
Delta			162.3	-132.4	143.8	-209.8	177.3	-208	185.5	-255	190.1	-221.4	157.5	-231.3	221.1	-215
02170512	-64.3	43.3	-84.6	153.6	-73.3	241	-47.4	246.6	-75.9	243.7	-77.8	241.5	-82.2	197.5	-67.8	231.4
Delta			-20.3	110.3	-9	197.7	16.9	203.3	-11.6	200.4	-13.5	198.2	-17.9	154.2	-3.5	188.1
02180500	-278.8	150.5	-113.1	184.6	-15.9	138.5	0.3	200.1	-15.6	249	-29.7	328	-16.9	325.7	1.8	287.9
Delta			165.7	34.1	262.9	-12	279.1	49.6	263.2	98.5	249.1	177.5	261.9	175.2	280.6	137.4
02180512	-220.3	514.7	-79.1	354.4	-40.1	279	4.2	241.9	-92.9	275.5	-112.5	240.5	-92.7	344.2	-63.7	245.2
Delta			141.2	-160.3	180.2	-235.7	224.5	-272.8	127.4	-239.2	107.8	-274.2	127.6	-170.5	156.6	-269.5
Valid Time																
04040500	176.4	90.5	184.6	48.2												
Delta			8.2	-42.3												
04040512	116.5	-100.3	146.8	-124.8	73.2	-59.5										
Delta			30.3	-24.5	-43.3	40.8										
04050500	269.4	-172	83.7	-58.1	102.3	-125.8	73	-102.2								
Delta			-185.7	113.9	-167.1	46.2	-196.4	69.8								
04050512	-73.3	-45.3	-194.9	61.8	-54	-56.3	-25.7	-45.8	-53.6	-16.5						
Delta			-121.6	107.1	19.3	-11	47.6	-0.5	19.7	28.8						
04060500	165.4	103.5	5.8	126.1	-27.7	107.3	15	63.3	27.9	15.4	24.2	59				
Delta			-159.6	22.6	-193.1	3.8	-150.4	-40.2	-137.5	-88.1	-141.2	-44.5				
04060512	55.5	104.1	-53.1	112	-51.4	127.4	-25.2	121.3	-45.6	112.3	-11	-10.9	-6.6	89.7		
Delta			-108.6	7.9	-106.9	23.3	-80.7	17.2	-101.1	8.2	-66.5	-115	-62.1	-14.4		
04070500	140	92.3	112.4	168.9	100.4	48.4	50.2	61.7	109.9	55	120.2	34.5	143.5	-13.5	133.5	72.7
Delta			-27.6	76.6	-39.6	-43.9	-89.8	-30.6	-30.1	-37.3	-19.8	-57.8	3.5	-105.8	-6.5	-19.6
04070512	-65.4	99.7	94.5	111.8	100.4	92.4	102.6	1.5	142.3	8.5	90	-62.5	74.9	-31.1	74.1	-50.4
Delta			159.9	12.1	165.8	-7.3	168	-98.2	207.7	-91.2	155.4	-162.2	140.3	-130.8	139.5	-150.1
04080500	293.1	-219.4	167.6	-175.7	163.4	-179.4	158.5	-158.1	134.3	-237.9	176.7	-209	169.9	-220.7	173.5	-116.5
Delta			-125.5	43.7	-129.7	40	-134.6	61.3	-158.8	-18.5	-116.4	10.4	-123.2	-1.3	-119.6	102.9
04080512	51.9	-16.3	54.3	-87.3	53.4	-166.7	23.7	-150.9	58.4	-148.6	30.9	-198.6	54.8	-208.8	52.2	-228.4
Delta			2.4	-71	1.5	-150.4	-28.2	-134.6	6.5	-132.3	-21	-182.3	2.9	-192.5	0.3	-212.1

Table 12. Comparison of chute agent outputs using Vandenberg 81km forecast and measured winds to 300 meters above ground level.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSION

Forecast winds are a useful planning tool to get a first look of the atmosphere and a ballpark expectation of what the conditions in the area are like. The resolution of data needed depends on the type of planning needed. Although it would be most beneficial to have the high-resolution data available to the forecast extent needed, providing that data is subject to time/bandwidth difficulty and higher costs.

For accuracy it is best to use data in real time and make decisions based on that information. Doppler weather systems, tied to GPS and linked to troops via high bandwidth connectivity, provide the best answer at the highest costs. Global models with concise data sets used over limited-bandwidth connections can also provide reasonable answers with high availability at much lower costs.

Future work could examine a more flexible analysis system to see if there were ways to minimize temporal smoothing. It may be possible to use past performance of models for the area to develop error-detection schemes and apply them forward through the forecast to avoid compounded errors that occur in temporal smoothing. At least it should be possible to develop a level of confidence in the forecast fields used for a particular area. Such indicators could indicate to the decision makers how much variability there may be in categorizing the weather impact.

A. AGENTS

Agents are beneficial for data extraction and developing knowledge from raw data. Their modularity allows you to develop complex planning or decision-making machines with a large degree of flexibility. As was demonstrated, when you have a conceptual error in an agent, you can remove the module and replace the logic. These qualities make agents adaptable to forecasting weather or oceanographic events. Forecasting is a complex set of repeatable tasks, and a good candidate for a modular forecasting-agent solution, one that can start with specific single-element forecasts and build to a complex system of logic to provide fast, accurate, and complete forecasts over

time. Additionally, an agent can provide the ability to develop forecasts, while building in a confidence index for forecast locations directly during the forecast modeling phase.

Overall this project demonstrates a quantifiable time savings in developing an automated process for evaluating a large system of specific types of data. The long term benefits increase over time, especially for repetitive forecasting requirements that can be easily quantified, such as winds over a region that fall within a certain range, or the examination of a region for a threshold amount of precipitation. Specifically searching areas where statistically long range forecast have a proven record of reliability in making accurate forecasts would benefit most from this method.

It is my opinion that, agents without a doubt have shown that it is possible to generate automated forecast systems, that can reliably analyze, and with the right types of evaluation rules embedded within them, even forecast weather events with a high degree of accuracy; at least with an accuracy on par with the specific weather model used to begin the forecast process.

B. IMPACT OF MODEL RESOLUTION

In studying the impact of data resolution and sensitivities on tactical decision aids, it can be seen that it depends specifically on the aid type. In the case of my bomb agent, data resolution has little impact on this less sensitive type of aid. Increasing the size and mass to the magnitude of the 1000lb and 2000lb bombs would likely make it less sensitive to the resolution. For the chute agent, data resolution had a moderate effect on this more sensitive aid. The winds could have such a large effect on the outcome of the agent that it can be concluded that only real-time winds and real-time guidance systems could compensate for parachuting cargo pallets wind sensitivity. But this doesn't make the planning data useless. The data could be useful to minimize weather effects on the mission. We can see from experiments that higher data resolution may have more effect on forecasting other weather parameters, like localized precipitation or visibility, and even when those solutions are not perfect, we can use the information to give baseline expectations and still maximize impacts.

Probably the most important discovery from the use of agents to examine model resolution is the discovery that temporal resolution is more important. With the phasing of a storm being crucial to the intensity and directionality of the winds, it can be stated that the better the temporal resolution of model, the more accurately it can reflect a change in state of the atmosphere. If those changes are well-phased between the model and the actual event, then the accuracy and confidence of the forecast is enhanced.

THIS PAGE INTENTIONALLY LEFT BLANK

BIBLIOGRAPHY

Blevins, R.D., *Applied Fluid Dynamics Handbook.* Krieger Publishing, 1992.

Gruntonline.com, "Fire Support Coordination: General Air Munitions Data," 2004. Available online at www.gruntonline.com/USForces/U.S. Artillery/arty13d.htm.

Jane's Information Group, "Joint Direct Attack Munition (JDAM): GBU-31, GBU-32, GBU-38," 2004. Available online at www.janes.com.

Kuethe, A.M., Chow, C.Y., *Foundations of Aerodynamics: Bases of Aerodynamic Design, 5th Edition.* John Wiley & Sons, Inc., 1998.

National Centers for Environmental Prediction, "Operational North American Meso: Output Grid Coverage," 2005. Available online at www.emc.ncep.noaa.gov /mmb/namgrids/.

Petrotechnical Open Standards Consortium, "Projections and Projected Coordinate Systems: Lambert Conic Conformal," 2007. Available online at www.posc.org/Epicentre.2_2/DataModel/ExamplesofUsage/eu_cs34e.html.

Press, A., "Glossary of Meteorology," retrieved 2007 from amsglossary.allenpress.com/glossary.

Stull, R.B., *An Introduction to Boundary Layer Meteorology*, Springer Publishing, 1988.

Wikipedia.org, "Main Page", retrieved 2007 from en.wikipedia.org/ wiki/Main_Page.

Wolfram Research, "Elliptic Integral of the Second Kind," 2005. Available online at mathworld.wolfram.com/EllipticIntegraloftheSecondKind.html.

World Meteorological Organization, "Part II: A Guide to the Code Form FM 92-IX Ext. GRIB Edition 1," 2005. Available online at www.wmo.ch/web/www/ WDM/Guides/Guide-binary-2.html.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX 1. AGENT INTERACTIONS

As was mentioned in Chapter I, agents are valuable elements to knowledge extraction in an automated data-retrieval system. Interoperability determines the output of the automated extraction of knowledge and necessitates the modularity of agents and their design. By making each agent modular, an automated knowledge-extraction system gains flexibility and extensibility, which means that an agent module can be modified, removed, or rearranged, or another agent can be added altogether to produce additional desired outputs. In addition, a similar system producing different knowledge entirely could be built by modifying the agent interactions. A self-modifying set of agents could produce specific knowledge on demand.

A. DATA COLLECTION

First this system must collect the meteorological data necessary to meet the end goal of comparing forecast wind values to real-world measured winds. While a pure agent system could have been developed to locate multiple sources for both forecast and measured winds, in the interest of time I chose to locate only two sources, one for measure RAOB's and one for forecast data. This approach is comprised of a time service, an FTP service, and an interface to manipulate file names. My primary source of data was from the National Centers for Environmental Prediction's public FTP server. After some research into how they rewrote their file names for each new forecast model run, I set to work developing the interface to manipulate the file names based on the time of the day. I interfaced the file name generator with the FTP service, and used a "cron" job to routinely download forecast data sets twice a day. Every twelve hours I was accumulating approximately one gigabyte of data, so I chose to archive the files to CD's. This was a cheap and efficient storage solution to long term archival of my research data.

The next step was to get actual wind measurements. In researching data availability, I found the University of Wyoming's Atmospheric Sciences department kept

a long term archive of data on line, with a Web interface that allowed me to download data for a specific period of time. I did this download manually and added the sets to my CD archives as I created them.

B. GRID EXTRACTIONS

The forecast data retrieved from NCEP's servers is in a WMO-specified format for gridded binary data, GRIB. To work with this data as a grid set, it was necessary for an agent to decode and format the data, but due to its complexity and the time I had to work with, I would need to add a decoding module into my decoding agent. Because of the graphical and scientific nature of this project, I had already chosen MATLAB as my modeling language, so I did Internet research for a MATLAB decoder for WMO GRIB data. I located only one freely available resource on the Internet, read_grib provided by Brian Blanton of the Marine Sciences Department of UNC Chapel Hill, NC. It proved to be quite a useful and powerful tool for decoding, but as with all free software it did have some discrepancies to correct.

To the read_grib source I made necessary modification to stabilize the software, and then set to wrapping the code with my data extraction agent. In my first attempt to extract data I realized I had one severe problem: the geodetic format that NCEP provides could not be simply decoded to latitude and longitude. The provided output in Lambert Conformal projection gave only the base coordinates for the lower left hand corner of the grid. Data after that was spaced on a grid of fixed distances, dependent on the resolution of the model. For example the 12km resolution model had grid outputs spaced 12km apart. To match the grid to actual latitude and longitude coordinates for a wind profile meant I had to go back and create a transformation. After a week of scouring the Internet for information, I located enough sources to begin developing a transformation. I used WMO information to determine that the 12km spacing for meridional spacing was measured at the central parallel of the grids. That indicated to me that if I computed the Earth's radius at that latitude, and used the 12km grid space, I could extract the longitude for every column of the grid.

The latitude was not so simple. From Wolfram Research's Math World web site (mathworld.worfram.com) I found a solution to compute the path length along an ellipse. It however looked to be computationally expensive. I had to weight that expense against the accuracy I required. Because my best resolution of winds was at 12km, I could only expect to be able to interpolate wind data for geographic locations between grid points. I chose to develop a less computationally expensive means to calculate latitude on an ellipse that would give me accuracy within a few meters on a par with that of a GPS locater. I chose an iterative method to estimate elliptical arc length in degrees, and converted that back to meters. By using an error estimate of less than one meter, I was able to generate latitude coordinates for my data grids that remained well within current GPS accuracy standards.

With the modules to generate a mesh grid of latitude and longitude to correspond to the data grids, I could extract data from my data sets. I took an initial inventory to determine which sets of wind data were available for extraction. With the appropriate list of parameters, I built an automate agent to extract the wind data from the GRIB data sets, and bilinear interpolation from that level what the winds would be at my specific target location. The wind data from that level was input to a profile matrix of elevation and wind components. When every wind level was extracted from the data set between the ground and 5000 meters, the profile was written to an output directory capturing the location and Zulu time of the wind profile, and forecast time of the model for unique reference and statistics purposes.

I also developed another simple agent in MATLAB to assist with the profiling of the winds from actual measurements recorded in the downloaded RAOB data. This proved to be a quick profile generator to take out the necessary wind inputs and generate component values before storing them in a similar profile format to the forecast wind data.

C. SIMULATION INPUTS

I set up an automated simulation program to run both a wind and bomb profile with all the wind sets available in one source directory. I only worked with small sets of

wind profiles to make tracking and data collection manageable. A more sophisticated agent-based system could automate the daily downloading, generating and simulation of wind profiles, and process a type of parameter searches for actual mission profiling.

D. STATISTICS ANALYSIS

My final set of agents did statistical analysis. These were straightforward agents which did a large amount of work with little human interaction. They wrapped up my profiles neatly, and generated graphics, on data for long-term measured averages, forecast averages, and a variety of other statistical analysis. Statistical agents can be added, removed or rewritten as necessary to meet analysis requirements.

APPENDIX 2. PRIMARY CODE SAMPLES (MATLAB FORMAT)

A. BOMBAGENT.M CODE

```
function bombagent(inMass,inLength,inMeanD,inCDaxial,inCDXaxial,inAlt, ...
   inSpeed,inDir,inAOA,inArg);

%Primary Agent code to enact the bombmodel research with input winds.  This
%module is called from the GUI code and implements the fall model against
%either a fixed set of winds or a set of profiles loaded in the profiles
%directory by the wind profile agent.

if (strcmp(inArg,'fixed'))
   %init research variables
   %windspeed = [0 1 2 5 10 20 50 100];
   windspeed = 20;
   %winddir = [0 45 90];
   winddir = 45;

   for m=1:length(winddir)
      for n=1:length(windspeed)
         bombfallmodel(inAlt,inSpeed,inAOA,inDir,inMass,inLength,inMeanD, ...
         inCDaxial,inCDXaxial,windPcreater(inAlt,windspeed(n),winddir(m)));
         %label and output the figures
         fl = ['run_' int2str(n) '_windspeed_' int2str(windspeed(n)) ...
            '_winddir_' int2str(winddir(m)) ];
         tstring = [fl '_1'];
         figure(1);
         print('-dtiff',tstring);
         tstring = [fl '_2'];
         figure(2);
         print('-dtiff',tstring);
         tstring = [fl '_3'];
         figure(3);
         print('-dtiff',tstring);
         tstring = [fl '_4'];
         figure(4);
         print('-dtiff',tstring);
         close all
      end %end windspeed loop
   end %end windir loop
else
   dircon = dir('./profiles');
   numentries = length(dircon);
   runnum = 1;
```

```
    for n=3:numentries
        %load a wind profile file
        fnam = ['./profiles/' dircon(n).name];
        inpro = dlmread(fnam);

        %Original try to build a 3d wind field from the profile
        %VWindIn = windprofile(inpro,inAlt);
        %RAWIN INPUT FROM FILE
        %VWindIn = wind2profile([inpro(:,2),inpro(:,8),inpro(:,7)],inAlt);
        %Model INPUT FROM GENERATED PROFILE 12km height limited to 1400m
        %Others to 4800m
        VWindIn = wind3profile(inpro,inAlt);

        %run the bomb agent
        bombfallmodel(inAlt,inSpeed,inAOA,inDir,inMass,inLength,inMeanD, ...
            inCDaxial,inCDXaxial,VWindIn);
        %output the figures
        fl = ['run_' int2str(runnum) '_windprofile_' dircon(n).name];
        tstring = [fl '_1'];
        figure(1);
        print('-dtiff',tstring);
        tstring = [fl '_2'];
        figure(2);
        print('-dtiff',tstring);
        tstring = [fl '_3'];
        figure(3);
        print('-dtiff',tstring);
        tstring = [fl '_4'];
        figure(4);
        print('-dtiff',tstring);
        close all
        runnum = runnum + 1;
    end
end
```

B. BOMBFALLMODEL.M CODE

```
function bombfallmodel(inAlt,inSpeed,inAngle,inDir,inMass, ...
    inLength,inMeanD,inCDaxial,inCDXaxial, inForcing)

% Complex Bomb Fall Model version 2
% Includes 3D plotting routine

% Includes gravity acceleration in the Z direction.
% Includes drag accelerations in the axial and cross
% axial directions.
```

```
% Includes wind effects from forcing

%Inputs:
%inMass - mass of the palette at release kg
%inLength - length of the bomb in m.
%inMeanD - avearage diameter of the bomb for surface area in m.
%inCDaxial - bomb axial drag coefficient unit less.
%inCDXaxial - bomb cross axial drag coeff. unit less.
%inAlt - release altitude in meters.
%inSpeed - initial velocity of the palette in m/s.
%inDir - initial compass direction of travel in deg.
%inAngle - angle of attack in degrees in m.
%inForcing - the wind forcing file.

%-----------------------------------------------------------------------
%SETUP VARIABLES
%-----------------------------------------------------------------------
% Velocity - Vcomp = ordered triple x, y, z vector
% Acceleration - Acomp = ordered triple ax, ay, az
% Direction of Attack - Dangle compass direction
% Position Pt = ordered 4 by N matrix for t,x,y,z vector

Hangle = inAngle/180 * pi; %convert to radians
Dangle = inDir/360 * 2 * pi; % y ordinate direction
CSArea = inMeanD * inLength;
SurfArea = pi * inMeanD * inLength;
Vtotal = inSpeed;

%Vectorize the velocity components
Vcomp = Vtotal * [sin(Dangle)*cos(Hangle), ...
   cos(Dangle)*cos(Hangle), sin(Hangle)]';
%Set initial conditions of 4D position tracer
Pt = [0,0,0,inAlt]';
height = inAlt;

%-----------------------------------------------------------------------
%BEGIN STEP CALCULATIONS TO BALANCE FORCES
%-----------------------------------------------------------------------

%CFL condition for stability is C dx / dt < 1
%Assuming dx to be 1 meter and C is Vtotal then
% dt < 1/Vtotal

stepindex = 1;
```

```
while (height > 0)
   %--------------------------------
   %ESTABLISH WIND RELATIONSHIP
   %--------------------------------
   lvlWindV = inForcing(2:4,ceil(height));
         WRelVel = Vcomp(:,stepindex) - lvlWindV;
   WRelVmag = sqrt(WRelVel(1)^2 + WRelVel(2)^2 + WRelVel(3)^2);
   dt = 1/WRelVmag;

         stepindex = stepindex + 1;

         %Change computation of new time to allow for variable
         %time step because relative velocity changes with gravity
   %and wind changes
         newTime = dt + Pt(1,stepindex - 1);

         %Calculate body directions
   %Angle to horizon flat earth assumption
   relThetaHorizI = asin(WRelVel(3)/WRelVmag);
   relMotionUnitI = WRelVel/WRelVmag;
   ZnormI = sqrt(WRelVel(1)^2 + WRelVel(2)^2) * tan(relThetaHorizI + pi/2);
   relMotionNormI = [WRelVel(1) WRelVel(2) ZnormI]' / ...
         sqrt(WRelVel(1)^2 + WRelVel(2)^2 + ZnormI^2);
   relMotionCrossI = cross(relMotionUnitI,relMotionNormI);


   %---------------------------------
   %ESTABLISH DRAG EFFECTS
   %---------------------------------
   %Calculate Drag forces of Motion, for Cross Axial Winds, and
   % Along Axis winds.

   relUnitFlowI = dot(WRelVel,relMotionUnitI);
   relNormFlowI = dot(WRelVel,relMotionNormI);
   relCrossFlowI = dot(WRelVel,relMotionCrossI);

   %1. Forward forecast the final relative velocity including drag effects
   DBI = dragforce([relUnitFlowI relNormFlowI relCrossFlowI],height, ...
      [inCDaxial inCDXaxial inCDXaxial], [SurfArea CSArea CSArea]);
   Dtotal = DBI(1)*-relMotionUnitI + DBI(2)*-relMotionNormI + ...
         DBI(3)*-relMotionCrossI;
         ACI = (Dtotal/inMass)+[0,0,-9.81]';
         VRelF = WRelVel + ACI*dt;

   %2. Backstep the velocity forecast with the reduced drag.
```

```
%New orientation

VRelMagF = sqrt(VRelF(1)^2 + VRelF(2)^2 + VRelF(3)^2);
relThetaHorizF = asin(VRelF(3)/VRelMagF);
relMotionUnitF = VRelF/VRelMagF;
ZnormF = sqrt(VRelF(1)^2 + VRelF(2)^2) * tan (relThetaHorizF + pi/2);
relMotionNormF = [VRelF(1) VRelF(2) ZnormF]' / ...
        sqrt(VRelF(1)^2 + VRelF(2)^2 + ZnormF^2);
relMotionCrossF = cross(relMotionUnitF,relMotionNormF);
%New flow and drag values
relUnitFlowF = dot(VRelF,relMotionUnitF);
relNormFlowF = dot(VRelF,relMotionNormF);
relCrossFlowF = dot(VRelF,relMotionCrossF);

DBF = dragforce([relUnitFlowF relNormFlowF relCrossFlowF],height, ...
    [inCDaxial inCDXaxial inCDXaxial], [SurfArea CSArea CSArea]);
Dtotal = DBF(1)*-relMotionUnitF + DBF(2)*-relMotionNormF + ...
        DBF(3)*-relMotionCrossF;
        ACF = (Dtotal/inMass)+[0,0,-9.81]';
%New Starting Velocity
        VRelI = VRelF - ACF*dt;

        %3. Assuming linear drag changes because of the small time step,
% Average the initial velocities and recompute the corrected drag.
VRelAv = (WRelVel + VRelI)/2;
%New orientation
VRelMagAv = sqrt(VRelAv(1)^2 + VRelAv(2)^2 + VRelAv(3)^2);
relThetaHorizAv = asin(VRelAv(3)/VRelMagAv);
relMotionUnitAv = VRelAv/VRelMagAv;
ZnormAv = sqrt(VRelAv(1)^2 + VRelAv(2)^2) * tan (relThetaHorizAv + pi/2);
relMotionNormAv = [VRelAv(1) VRelAv(2) ZnormAv]' / ...
        sqrt(VRelAv(1)^2 + VRelAv(2)^2 + ZnormAv^2);
relMotionCrossAv = cross(relMotionUnitAv,relMotionNormAv);
%New flow and drag values
relUnitFlowAv = dot(VRelAv,relMotionUnitAv);
relNormFlowAv = dot(VRelAv,relMotionNormAv);
relCrossFlowAv = dot(VRelAv,relMotionCrossAv);

DBA = dragforce([relUnitFlowAv relNormFlowAv relCrossFlowAv],height, ...
    [inCDaxial inCDXaxial inCDXaxial], [SurfArea CSArea CSArea]);
Dtotal = DBA(1)*-relMotionUnitAv + DBA(2)*-relMotionNormAv + ...
        DBA(3)*-relMotionCrossAv;
        ACA = (Dtotal/inMass)+[0,0,-9.81]';
%Final Solution Velocity
        VRelA = WRelVel + ACA*dt;
```

```
        %Remove Relative wind effects to cast to actual earth motions
        VActual = VRelA + lvlWindV;

        %Calculate the position with time averaged velocity
        newX = (VActual(1)*dt) + Pt(2,stepindex-1);
        newY = (VActual(2)*dt) + Pt(3,stepindex-1);
        newZ = (VActual(3)*dt) + Pt(4,stepindex-1);

        Pt(:,stepindex) = [newTime,newX,newY,newZ]';
        Vcomp(:,stepindex)=VActual;
        Vtotal = sqrt(VActual(1)^2 + VActual(2)^2 + VActual(3)^2);

        height = newZ;

end %while loop

%Plotting the figures
%Sanity check values
%Vcomp(:,stepindex)
%Pt(:,stepindex)

plotmodelws;
```

C. CHUTEAGENT.M CODE

```
function chuteagent(inMass,inSLength,inChuteD,inCDaxialC,inCDXaxialC,inAlt, ...
   inSpeed,inDir,inSHeight,inCDaxialB,inCDXaxialB,inArg);

%Primary Agent code to enact the chutemodel research with input winds.  This
%module is called from the GUI code and implements the fall model against
%either a fixed set of winds or a set of profiles loaded in the profiles
%directory by the wind profile agent.

if (strcmp(inArg,'fixed'))
   %init research variables
   %windspeed = [0 1 2 5 10 20 50 100];
   %winddir = [0 45 90];
   windspeed = 20;
   winddir = 45;

   for m=1:length(winddir)
      for n=1:length(windspeed)
         chutefallmodel(inMass,inSLength,inChuteD,inCDaxialC,inCDXaxialC,inAlt, ...
            inSpeed,inDir,inSHeight,inCDaxialB,inCDXaxialB,...
            windPcreater(inAlt,windspeed(n),winddir(m)));
```

```
        %label and output the figures
        fl = ['run_' int2str(n) '_windspeed_' int2str(windspeed(n)) ...
          '_winddir_' int2str(winddir(m)) ];
        tstring = [fl '_1'];
        figure(1);
        print('-dtiff',tstring);
        tstring = [fl '_2'];
        figure(2);
        print('-dtiff',tstring);
        tstring = [fl '_3'];
        figure(3);
        print('-dtiff',tstring);
        tstring = [fl '_4'];
        figure(4);
        print('-dtiff',tstring);
        close all
      end %end windspeed loop
    end %end windir loop
  else
    dircon = dir('./profiles');
    numentries = length(dircon);
    runnum = 1;
    for n=3:numentries
      %load a wind profile file
      fnam = ['./profiles/' dircon(n).name];
      inpro = dlmread(fnam);

      %Original try to build a 3d wind field from the profile
      %VWindIn = windprofile(inpro,inAlt);
      %RAWIN INPUT FROM FILE
      %VWindIn = wind2profile([inpro(:,2),inpro(:,8),inpro(:,7)],inAlt);
      %Model INPUT FROM GENERATED PROFILE 12km height limited to 1400m
      %Others to 4800m
      VWindIn = wind3profile(inpro,inAlt);

      %run the bomb agent
      chutefallmodel(inMass,inSLength,inChuteD,inCDaxialC,inCDXaxialC,inAlt, ...
          inSpeed,inDir,inSHeight,inCDaxialB,inCDXaxialB,VWindIn);
      %output the figures
      fl = ['run_' int2str(runnum) '_windprofile_' dircon(n).name];
      tstring = [fl '_1'];
      figure(1);
      print('-dtiff',tstring);
      tstring = [fl '_2'];
      figure(2);
```

```
        print('-dtiff',tstring);
        tstring = [fl '_3'];
        figure(3);
        print('-dtiff',tstring);
        tstring = [fl '_4'];
        figure(4);
        print('-dtiff',tstring);
        close all
        runnum = runnum + 1;
    end
end
```

D. CHUTEFALLMODEL.M CODE

```
function chutefallmodel(inMass,inSLength,inChuteD,inCDaxialC,inCDXaxialC,inAlt, ...
            inSpeed,inDir,inSHeight,inCDaxialB,inCDXaxialB,inForcing)
% Complex Chute Fall Model
% Includes 3D plotting routine

% Includes gravity acceleration in the Z direction.
% Includes drag accelerations in the axial and cross
% axial directions.
% Includes wind effects from forcing

%Inputs:
%inMass - mass of the palette at release kg
%inSLength - length of a side of the square palette in m.
%inChuteD - diameter of the drag chute in m.
%inCDaxialC - chute axial drag coefficient unit less.
%inCDXaxialC - chute cross axial drag coeff. unit less.
%inAlt - release altitude in meters.
%inSpeed - initial velocity of the palette in m/s.
%inDir - initial compass direction of travel in deg.
%inSHeight - height of side of palette in m.
%inCDaxialB - drag coefficient for flow on the palette down the axis.
%inCDXaxialB - drag coefficient for flow on the palette across the axis.
%inForcing - the wind forcing file.

%-----------------------------------------------------------------------
%SETUP VARIABLES
%-----------------------------------------------------------------------
% Velocity - Vcomp = ordered triple x, y, z vector
% Acceleration - Acomp = ordered triple ax, ay, az
% Position Pt = ordered 4 by N matrix for t,x,y,z vector

Dangle = inDir/360 * 2 * pi; % y ordinate direction
```

```
SideArea = inSLength * inSHeight;
BottomArea = inSLength^2;
Vtotal = inSpeed;

%Vectorize the velocity components
Vcomp = Vtotal * [sin(Dangle), cos(Dangle), 0]';
%Set initial conditions of 4D position tracer
Pt = [0,0,0,inAlt]';
height = inAlt;

%----------------------------------------------------------------------
%BEGIN STEP CALCULATIONS TO BALANCE FORCES
%----------------------------------------------------------------------

%CFL condition for stability is C dx / dt < 1
%Assuming dx to be 1 meter and C is Vtotal then
% dt < 1/Vtotal

stepindex = 1;

while (height > 0)
   %--------------------------------
   %ESTABLISH WIND RELATIONSHIP
   %--------------------------------
   lvlWindV = inForcing(2:4,ceil(height));
         WRelVel = Vcomp(:,stepindex) - lvlWindV;
   WRelVmag = sqrt(WRelVel(1)^2 + WRelVel(2)^2 + WRelVel(3)^2);
   dt = 1/WRelVmag;

         stepindex = stepindex + 1;

         %Change computation of new time to allow for variable
         %time step because relative velocity changes with gravity
   %and wind changes
         newTime = dt + Pt(1,stepindex - 1);
   %set delay for chute to open
   if (newTime < 3.0)
      chuteDaxial = inCDaxialC * (newTime/3.0);
      chuteDXaxial = inCDXaxialC * (newTime/3.0);
      ChuteArea = pi * (inChuteD * newTime/3.0)^2 * 0.25;
   else
      chuteDaxial = inCDaxialC;
      chuteDXaxial = inCDXaxialC;
      ChuteArea = pi * inChuteD^2 * 0.25;
   end
```

```
    %Calculate body directions
%Angle to horizon flat earth assumption
relThetaHorizI = asin(WRelVel(3)/WRelVmag);
relMotionUnitI = WRelVel/WRelVmag;
ZnormI = sqrt(WRelVel(1)^2 + WRelVel(2)^2) * tan (relThetaHorizI + pi/2);
relMotionNormI = [WRelVel(1) WRelVel(2) ZnormI]' / …
    sqrt(WRelVel(1)^2 + WRelVel(2)^2 + ZnormI^2);
relMotionCrossI = cross(relMotionUnitI,relMotionNormI);


%---------------------------------
%ESTABLISH DRAG EFFECTS
%---------------------------------
%Calculate Drag forces of Motion, for Cross Axial Winds, and
% Along Axis winds.

relUnitFlowI = dot(WRelVel,relMotionUnitI);
relNormFlowI = dot(WRelVel,relMotionNormI);
relCrossFlowI = dot(WRelVel,relMotionCrossI);

%1. Forward forecast the final relative velocity including drag effects
DCI = dragforce([relUnitFlowI relNormFlowI relCrossFlowI],height, ...
  [chuteDaxial chuteDXaxial chuteDXaxial], [ChuteArea ChuteArea ChuteArea]);
DBI = dragforce([relUnitFlowI relNormFlowI relCrossFlowI],height, ...
  [inCDaxialB inCDXaxialB inCDXaxialB], [BottomArea SideArea SideArea]);
Dtotal = ((DCI(1) + DBI(1))*-relMotionUnitI) + ...
  ((DCI(2) + DBI(2))*-relMotionNormI) + ((DCI(3) + DBI(3))*-relMotionCrossI);
    ACI = (Dtotal/inMass)+[0,0,-9.81]';
    VRelF = WRelVel + ACI*dt;
    %2. Backstep the velocity forecast with the reduced drag.
%New orientation
VRelMagF = sqrt(VRelF(1)^2 + VRelF(2)^2 + VRelF(3)^2);
relThetaHorizF = asin(VRelF(3)/VRelMagF);
relMotionUnitF = VRelF/VRelMagF;
ZnormF = sqrt(VRelF(1)^2 + VRelF(2)^2) * tan (relThetaHorizF + pi/2);
relMotionNormF = [VRelF(1) VRelF(2) ZnormF]' / …
    sqrt(VRelF(1)^2 + VRelF(2)^2 + ZnormF^2);
relMotionCrossF = cross(relMotionUnitF,relMotionNormF);
%New flow and drag values
relUnitFlowF = dot(VRelF,relMotionUnitF);
relNormFlowF = dot(VRelF,relMotionNormF);
relCrossFlowF = dot(VRelF,relMotionCrossF);

DCF = dragforce([relUnitFlowF relNormFlowF relCrossFlowF],height, ...
```

```
   [chuteDaxial chuteDXaxial chuteDXaxial], [ChuteArea ChuteArea ChuteArea]);
DBF = dragforce([relUnitFlowF relNormFlowF relCrossFlowF],height, ...
   [inCDaxialB inCDXaxialB inCDXaxialB], [BottomArea SideArea SideArea]);
Dtotal = ((DCF(1) + DBF(1))*-relMotionUnitF) + ...
   ((DCF(2) + DBF(2))*-relMotionNormF) + ((DCF(3) + DBF(3))*-relMotionCrossF);
      ACF = (Dtotal/inMass)+[0,0,-9.81]';
%New Starting Velocity
      VRelI = VRelF - ACF*dt;

      %3. Assuming linear drag changes because of the small time step,
% Average the initial velocities and recompute the corrected drag.
VRelAv = (WRelVel + VRelI)/2;
%New orientation
VRelMagAv = sqrt(VRelAv(1)^2 + VRelAv(2)^2 + VRelAv(3)^2);
relThetaHorizAv = asin(VRelAv(3)/VRelMagAv);
relMotionUnitAv = VRelAv/VRelMagAv;
ZnormAv = sqrt(VRelAv(1)^2 + VRelAv(2)^2) * tan (relThetaHorizAv + pi/2);
relMotionNormAv = [VRelAv(1) VRelAv(2) ZnormAv]' / …
      sqrt(VRelAv(1)^2 + VRelAv(2)^2 + ZnormAv^2);
relMotionCrossAv = cross(relMotionUnitAv,relMotionNormAv);
%New flow and drag values
relUnitFlowAv = dot(VRelAv,relMotionUnitAv);
relNormFlowAv = dot(VRelAv,relMotionNormAv);
relCrossFlowAv = dot(VRelAv,relMotionCrossAv);

DCA = dragforce([relUnitFlowAv relNormFlowAv relCrossFlowAv],height, ...
   [chuteDaxial chuteDXaxial chuteDXaxial], [ChuteArea ChuteArea ChuteArea]);
DBA = dragforce([relUnitFlowAv relNormFlowAv relCrossFlowAv],height, ...
   [inCDaxialB inCDXaxialB inCDXaxialB], [BottomArea SideArea SideArea]);
Dtotal = ((DCA(1) + DBA(1))*-relMotionUnitAv) + ...
   ((DCA(2) + DBA(2))*-relMotionNormAv) + …
      ((DCA(3) + DBA(3))*-relMotionCrossAv);
      ACA = (Dtotal/inMass)+[0,0,-9.81]';
%Final Solution Velocity
      VRelA = WRelVel + ACA*dt;

      %Remove Relative wind effects to cast to actual earth motions
      VActual = VRelA + lvlWindV;

      %Calculate the position with time averaged velocity
      newX = (VActual(1)*dt) + Pt(2,stepindex-1);
      newY = (VActual(2)*dt) + Pt(3,stepindex-1);
      newZ = (VActual(3)*dt) + Pt(4,stepindex-1);

      Pt(:,stepindex) = [newTime,newX,newY,newZ]';
```

```
        Vcomp(:,stepindex)=VActual;
        Vtotal = sqrt(VActual(1)^2 + VActual(2)^2 + VActual(3)^2);

        height = newZ;

end %while loop

%Plotting the figures

plotmodelchute;
```

E. GRIDAGENT2.M CODE

```
function retcod = gridagent2();
% agent tool to process grib data stored on cd and create profiles for a
% specified geographic location

origin = eval('cd');
target = 'd:';  %data stored on cd rom on d-drive;
eval(strcat(['cd ', target]));

%list directory contents to create a parse list of files
fnames = dir;
for i=length(fnames):-1:1
   if ~isempty(findstr(fnames(i).name,'.txt'))
      fnames = fnames(1:i-1);
   end
end

%Begin parsing list for each file to process as a wind profile.
for flist = 1:length(fnames)
   recs=read_grib(fnames(flist).name,1,'ScreenDiag',0);
   gridid = recs(1).pds.GridID;

   datadate = filedate(fnames(flist).date);
   vdate = vtdate(recs(1).pds);

   if ~isempty(findstr(fnames(flist).name,'t00'))
      cycle = '00';
   else
      cycle = '12';
   end

   %Function call to locate the gridded data onto a geographic datum from
   %information inside the grib grid description file.
   gridpos = makegp(recs(1).gds);
```

```
%Profiles are for a fixed know location.  Later modifications can be
%made to create an input structure to produce for any location.
rnkinf = struct('rnklat', dms2deg(37,19,00), 'rnklon', ...
   dms2deg(-79,58,00), 'rnkele', 358);
vbginf = struct('vbglat', dms2deg(34,39,00), 'vbglon', ...
   dms2deg(-120,34,00), 'vbgele', 112);

%In this switch statement the id field of the grid flags the resolution
%of the model.  Each id is for a different model with different model
%resolution, grid numbers, and products as defined my NCEP's model
%page.  Thus each case requires a module to handle it's profiling task
%accurately.  MORE AI work is required here to make the agents more
%autonomous WRT the input fields.
switch gridid
   case 218
      res = '12';
      wp = profile12(fnames(flist).name,gridpos,rnkinf,vbginf);
      sfrnkn = [res 'rnkprof' datadate cycle 'vt' vdate '.prf'];
      sfvbgn = [res 'vbgprof' datadate cycle 'vt' vdate '.prf'];
      pro1 = wp.rnk;
      pro2 = wp.vbg;
      eval( 'cd c:\thesis\profiles\12kmmodel\');
      dlmwrite(sfrnkn, pro1, '\t');
      dlmwrite(sfvbgn, pro2, '\t');
   case 221
      res = '32';
      wp = profile32(fnames(flist).name,gridpos,rnkinf,vbginf);
      sfrnkn = [res 'rnkprof' datadate cycle 'vt' vdate '.prf'];
      sfvbgn = [res 'vbgprof' datadate cycle 'vt' vdate '.prf'];
      pro1 = wp.rnk;
      pro2 = wp.vbg;
      eval( 'cd c:\thesis\profiles\32kmmodel\');
      dlmwrite(sfrnkn, pro1, '\t');
      dlmwrite(sfvbgn, pro2, '\t');
   case 211
      res = '81';
      wp = profile81(fnames(flist).name,gridpos,rnkinf,vbginf);
      sfrnkn = [res 'rnkprof' datadate cycle 'vt' vdate '.prf'];
      sfvbgn = [res 'vbgprof' datadate cycle 'vt' vdate '.prf'];
      pro1 = wp.rnk;
      pro2 = wp.vbg;
      eval( 'cd c:\thesis\profiles\81kmmodel\');
      dlmwrite(sfrnkn, pro1, '\t');
      dlmwrite(sfvbgn, pro2, '\t');
```

```
    otherwise
        disp(['Unrecognized Grid Resolution, Unable to process'])
  end
  eval(strcat(['cd ', target]));
  %clear all;
end

eval(['cd ' origin]);
retcod = 0;
%end gridagent


%%PRIVATE FUNCTIONS
function atm_profile = profile12(gribfn,posdat, rnkinf, vbginf);
% Process the predefined wind levels of the 12km file to create a profile
hvgridofint = [9, 33, 34, 35, 36, 38, 39, 40, 41, 42, 43, 44, 45, ...
    46, 47, 48, 49, 50, 51, 56, 57, 113, 114, 115];
hfgridofint = [16, 17];
fxlvofint = [10];

atm_profile = gpmprocess(gribfn, posdat, ...
        hvgridofint, hfgridofint, fxlvofint, rnkinf, vbginf);
%end profile12

% Process the predefined wind levels of the 32km file to create a profile
function atm_profile = profile32(gribfn,posdat, rnkinf, vbginf)
hvgridofint = [268, 273, 274, 278, 282, 283, 287, 291, 292,...
    296, 300, 301, 305, 309, 310, 314, 318, 319, 323, 327, 328, ...
    332, 336, 337, 341, 346, 347, 351, 355, 356, 360, 364, 365, ...
    369, 373, 374, 378, 382, 383, 387, 391, 392, 396, 402, 403, ...
    407, 411, 412, 416, 420, 421, 425, 429, 430, 434, 439, 440, ...
    444, 448, 449, 453, 458, 459];
hfgridofint = [13, 14, 555, 556, 558, 559, 561, 562, 564, 565, 567, ...
    568, 570, 571];
fxlvofint = [10, 914, 1524, 1829, 2134, 2743, 3658];

atm_profile = stdprocess(gribfn, posdat, ...
        hvgridofint, hfgridofint, fxlvofint, rnkinf, vbginf);
%end profile32

function atm_profile = profile81(gribfn,posdat, rnkinf, vbginf);
% Process the predetermined wind levels of the 32km file to create a profile
hvgridofint = [105, 108, 109, 110, 113, 114, 115, 118, 119, 120, 123, 124,...
    125, 128, 129, 130, 133, 134, 135, 138, 139, 140, 143, 144,...
    145, 148, 149, 150, 153, 154, 155, 158, 159];
```

```
hfgridofint = [12, 13];
fxlvofint = [10];

atm_profile = stdprocess(gribfn, posdat, ...
            hvgridofint, hfgridofint, fxlvofint, rnkinf, vbginf);
%end profile81

function apreturn = stdprocess(gribfn, posdat, ...
                 vargoi, fxgoi, fxlvoi, rnkinf, vbginf)
% This defines the standardize process to generate a profile, by extracting
% u and v velocity wind fields and geopotential heights directly from layer
% data inside the grib data file.

idx1sz = length(vargoi);
idx2sz = length(fxgoi);

rnkpro = [];
vbgpro = [];

[lonpoints,latpoints,varsponge]=size(posdat);

%Using a function call grid walk, I determine where inside the data grid
%the locations for Vandenberg and Roanoke are located.  (ie which two rows,
%and which two columns are the profile positions)  Once I have these
%locations I can interpolate for the wind speed between the grid cells.
rnkpoi = gridwalk(posdat,rnkinf.rnklon,rnkinf.rnklat);
vbgpoi = gridwalk(posdat,vbginf.vbglon,vbginf.vbglat);

%deternine the upper right hand corner for interpolation purposes
rnklatoi = rnkpoi(2);
rnklonoi = rnkpoi(1);
vbglatoi = vbgpoi(2);
vbglonoi = vbgpoi(1);

gridlats = posdat(:,:,1);
gridlons = posdat(:,:,2);

%set the interpolation values for the grid cells
rnkdlat = (rnkinf.rnklat-gridlats(rnklonoi,rnklatoi))/...
   (gridlats(rnklatoi-1)-gridlats(rnklonoi,rnklatoi));
rnkdlon = (rnkinf.rnklon-gridlons(rnklonoi,rnklatoi))/...
   (gridlons(rnklonoi-1)-gridlons(rnklonoi,rnklatoi));
vbgdlat = (vbginf.vbglat-gridlats(vbglonoi,vbglatoi))/...
   (gridlats(vbglatoi-1)-gridlats(vbglonoi,vbglatoi));
vbgdlon = (vbginf.vbglon-gridlons(vbglonoi,vbglatoi))/...
```

```
   (gridlons(vbglonoi-1)-gridlons(vbglonoi,vbglatoi));

%Begin processing records by reading the grib data and building profiles
%that contain the u wind speed v wind speed and geopotential height
gribrec = read_grib(gribfn,vargoi,'ScreenDiag',0);

for idx = 1:3:idx1sz
        profile1 = reshape(gribrec(idx).fltarray,lonpoints,latpoints)';
   profile2 = reshape(gribrec(idx+1).fltarray,lonpoints,latpoints)';
   profile3 = reshape(gribrec(idx+2).fltarray,lonpoints,latpoints)';

        rnkval1 = MyBI([[profile1(rnklatoi,rnklonoi-1),profile1(rnklatoi,rnklonoi)];...
             [profile1(rnklatoi-1,rnklonoi-1),profile1(rnklatoi-1, rnklonoi)]],...
          rnkdlon,rnkdlat);
        rnkval2 = MyBI([[profile2(rnklatoi,rnklonoi-1),profile2(rnklatoi,rnklonoi)];...
             [profile2(rnklatoi-1,rnklonoi-1),profile2(rnklatoi-1, rnklonoi)]],...
          rnkdlon,rnkdlat);
        rnkval3 = MyBI([[profile3(rnklatoi,rnklonoi-1),profile3(rnklatoi,rnklonoi)];...
             [profile3(rnklatoi-1,rnklonoi-1),profile3(rnklatoi-1, rnklonoi)]],...
          rnkdlon,rnkdlat);

        vbgval1 = MyBI([[profile1(vbglatoi,vbglonoi-1),profile1(vbglatoi,vbglonoi)];...
             [profile1(vbglatoi-1,vbglonoi-1),profile1(vbglatoi-1, vbglonoi)]],...
          vbgdlon,vbgdlat);
        vbgval2 = MyBI([[profile2(vbglatoi,vbglonoi-1),profile2(vbglatoi,vbglonoi)];...
             [profile2(vbglatoi-1,vbglonoi-1),profile2(vbglatoi-1, vbglonoi)]],...
          vbgdlon,vbgdlat);
        vbgval3 = MyBI([[profile3(vbglatoi,vbglonoi-1),profile3(vbglatoi,vbglonoi)];...
             [profile3(vbglatoi-1,vbglonoi-1),profile3(vbglatoi-1, vbglonoi)]],...
          vbgdlon,vbgdlat);

   rnkpro = [rnkpro;[rnkval1,rnkval2,rnkval3]];
   vbgpro = [vbgpro;[vbgval1,vbgval2,vbgval3]];
end
clear profile1;

%Continuing to build more profiles this time from u and v wind speeds this
%time at wind layers with pre set altitudes.
gribrec = read_grib(gribfn,fxgoi,'ScreenDiag',0);

for idx = 1:2:idx2sz
        profile2 = reshape(gribrec(idx).fltarray,lonpoints,latpoints)';
   profile3 = reshape(gribrec(idx+1).fltarray,lonpoints,latpoints)';

        rnkval1 = fxlvoi(ceil(idx/2)) + rnkinf.rnkele;
```

```
        rnkval2 = MyBI([[profile2(rnklatoi,rnklonoi-1),profile2(rnklatoi,rnklonoi)];...
            [profile2(rnklatoi-1,rnklonoi-1),profile2(rnklatoi-1, rnklonoi)]],...
          rnkdlon,rnkdlat);
        rnkval3 = MyBI([[profile3(rnklatoi,rnklonoi-1),profile3(rnklatoi,rnklonoi)];...
            [profile3(rnklatoi-1,rnklonoi-1),profile3(rnklatoi-1, rnklonoi)]],...
          rnkdlon,rnkdlat);

        vbgval1 = fxlvoi(ceil(idx/2)) + vbginf.vbgele;
        vbgval2 = MyBI([[profile2(vbglatoi,vbglonoi-1),profile2(vbglatoi,vbglonoi)];...
            [profile2(vbglatoi-1,vbglonoi-1),profile2(vbglatoi-1, vbglonoi)]],...
          vbgdlon,vbgdlat);
        vbgval3 = MyBI([[profile3(vbglatoi,vbglonoi-1),profile3(vbglatoi,vbglonoi)];...
            [profile3(vbglatoi-1,vbglonoi-1),profile3(vbglatoi-1, vbglonoi)]],...
          vbgdlon,vbgdlat);

   rnkpro = [rnkpro;[rnkval1,rnkval2,rnkval3]];
   vbgpro = [vbgpro;[vbgval1,vbgval2,vbgval3]];
end

%Sort by increasing altitude
rnkpro = myRevRow(sortrows(rnkpro));
vbgpro = myRevRow(sortrows(vbgpro));

%Remove the ground level altitude above Mean Sea Level from the
%geopotential height of the winds to get true layer heights above ground
%level.
rnkpro = [rnkpro(:,1)-rnkinf.rnkele,rnkpro(:,2),rnkpro(:,3)];
enddat = find(rnkpro(:,1) < 0);
if ~isempty(enddat)
   rnkpro = rnkpro(1:enddat-1,:);
end

vbgpro = [vbgpro(:,1)-vbginf.vbgele,vbgpro(:,2),vbgpro(:,3)];
enddat = find(vbgpro(:,1) < 0);
if ~isempty(enddat)
   vbgpro = vbgpro(1:enddat-1,:);
end

profile = struct('rnk',rnkpro,'vbg',vbgpro);

apreturn=profile;
%end stdprocess

function apreturn = gpmprocess(gribfn, posdat, ...
            vargoi, fxgoi, fxlvoi, rnkinf, vbginf)
```

```
% This defines the process to generate a profile, by extracting
% u and v velocity wind fields, along with temperature and mosture
% values to calculate geopotential heights when layer
% data is not available inside the grib data file.
idx1sz = length(vargoi);
idx2sz = length(fxgoi);

rnkpro = [];
vbgpro = [];

[lonpoints,latpoints,varsponge]=size(posdat);

%Using a function call grid walk, I determine where inside the data grid
%the locations for Vandenberg and Roanoke are located.  (ie which two rows,
%and which two columns are the profile positions)  Once I have these
%locations I can interpolate for the wind speed between the grid cells.
rnkpoi = gridwalk(posdat,rnkinf.rnklon,rnkinf.rnklat);
vbgpoi = gridwalk(posdat,vbginf.vbglon,vbginf.vbglat);

%deternine the upper right hand corner for interpolation purposes
rnklatoi = rnkpoi(2);
rnklonoi = rnkpoi(1);
vbglatoi = vbgpoi(2);
vbglonoi = vbgpoi(1);

gridlats = posdat(:,:,1);
gridlons = posdat(:,:,2);

%Set the interpolation values for the grid cells
rnkdlat = (rnkinf.rnklat-gridlats(rnklonoi,rnklatoi))/...
   (gridlats(rnklatoi-1)-gridlats(rnklonoi,rnklatoi));
rnkdlon = (rnkinf.rnklon-gridlons(rnklonoi,rnklatoi))/...
   (gridlons(rnklonoi-1)-gridlons(rnklonoi,rnklatoi));
vbgdlat = (vbginf.vbglat-gridlats(vbglonoi,vbglatoi))/...
   (gridlats(vbglatoi-1)-gridlats(vbglonoi,vbglatoi));
vbgdlon = (vbginf.vbglon-gridlons(vbglonoi,vbglatoi))/...
   (gridlons(vbglonoi-1)-gridlons(vbglonoi,vbglatoi));

%Begin processing records by reading the grib data and building profiles
%that contain the u wind speed v wind speed and geopotential height.  This
%process is unique in that grids are read in by set, to correlate the wind
%speeds, temperature and moisture to allow you to generate a geopotential
%height for a mid layer pressure where the wind values should occur.

sgribrec = read_grib(gribfn,vargoi,'ScreenDiag',0);
```

```
gribrec = [sgribrec(1:6),sgribrec(20),sgribrec(7:13),sgribrec(21),...
       sgribrec(14:19),sgribrec(22:24)];
clear sgribrec;

profile1 = reshape(gribrec(1).fltarray,lonpoints,latpoints)';
rnkpress = MyBI([[profile1(rnklatoi,rnklonoi-1),profile1(rnklatoi,rnklonoi)];...
         [profile1(rnklatoi-1,rnklonoi-1),profile1(rnklatoi-1, rnklonoi)]],...
         rnkdlon,rnkdlat)*0.01;
vbgpress = MyBI([[profile1(vbglatoi,vbglonoi-1),profile1(vbglatoi,vbglonoi)];...
         [profile1(vbglatoi-1,vbglonoi-1),profile1(vbglatoi-1, vbglonoi)]],...
         vbgdlon,vbgdlat)*0.01;

layertrack = 0;
for idx = 2:4:idx1sz-3
   layertrack = layertrack + 1;
         profile1 = reshape(gribrec(idx).fltarray,lonpoints,latpoints)';
   profile2 = reshape(gribrec(idx+1).fltarray,lonpoints,latpoints)';
   profile3 = reshape(gribrec(idx+2).fltarray,lonpoints,latpoints)';
   profile4 = reshape(gribrec(idx+3).fltarray,lonpoints,latpoints)';

         rnkval1 = MyBI([[profile1(rnklatoi,rnklonoi-1),profile1(rnklatoi,rnklonoi)];...
           [profile1(rnklatoi-1,rnklonoi-1),profile1(rnklatoi-1, rnklonoi)]],...
           rnkdlon,rnkdlat);
         rnkval2 = MyBI([[profile2(rnklatoi,rnklonoi-1),profile2(rnklatoi,rnklonoi)];...
           [profile2(rnklatoi-1,rnklonoi-1),profile2(rnklatoi-1, rnklonoi)]],...
           rnkdlon,rnkdlat);
         rnkval3 = MyBI([[profile3(rnklatoi,rnklonoi-1),profile3(rnklatoi,rnklonoi)];...
           [profile3(rnklatoi-1,rnklonoi-1),profile3(rnklatoi-1, rnklonoi)]],...
           rnkdlon,rnkdlat);
         rnkval4 = MyBI([[profile4(rnklatoi,rnklonoi-1),profile4(rnklatoi,rnklonoi)];...
           [profile4(rnklatoi-1,rnklonoi-1),profile4(rnklatoi-1, rnklonoi)]],...
           rnkdlon,rnkdlat);

         vbgval1 = MyBI([[profile1(vbglatoi,vbglonoi-1),profile1(vbglatoi,vbglonoi)];...
           [profile1(vbglatoi-1,vbglonoi-1),profile1(vbglatoi-1, vbglonoi)]],...
           vbgdlon,vbgdlat);
         vbgval2 = MyBI([[profile2(vbglatoi,vbglonoi-1),profile2(vbglatoi,vbglonoi)];...
           [profile2(vbglatoi-1,vbglonoi-1),profile2(vbglatoi-1, vbglonoi)]],...
           vbgdlon,vbgdlat);
         vbgval3 = MyBI([[profile3(vbglatoi,vbglonoi-1),profile3(vbglatoi,vbglonoi)];...
           [profile3(vbglatoi-1,vbglonoi-1),profile3(vbglatoi-1, vbglonoi)]],...
           vbgdlon,vbgdlat);
         vbgval4 = MyBI([[profile4(vbglatoi,vbglonoi-1),profile4(vbglatoi,vbglonoi)];...
           [profile4(vbglatoi-1,vbglonoi-1),profile4(vbglatoi-1, vbglonoi)]],...
           vbgdlon,vbgdlat);
```

```
    rnkeval = 6.11*exp((2.453*10^6/461)*(1/273-1/rnkval1))*rnkval2/100;
    vbgeval = 6.11*exp((2.453*10^6/461)*(1/273-1/vbgval1))*vbgval2/100;

    rnkwval = (0.622*rnkeval)/(rnkpress-(layertrack*30-15)-rnkeval);
    vbgwval = (0.622*vbgeval)/(vbgpress-(layertrack*30-15)-vbgeval);

    rnktv = rnkval1*(1+0.61*rnkwval);
    vbgtv = vbgval1*(1+0.61*vbgwval);

    rnkgpm = 287*rnktv*1/9.8*log(rnkpress/(rnkpress-(layertrack*30-15)));
    vbggpm = 287*vbgtv*1/9.8*log(vbgpress/(vbgpress-(layertrack*30-15)));

    rnkpro = [rnkpro;[rnkgpm,rnkval3,rnkval4]];
    vbgpro = [vbgpro;[vbggpm,vbgval3,vbgval4]];
end

profile1 = reshape(gribrec(idx1sz-2).fltarray,lonpoints,latpoints)';
profile2 = reshape(gribrec(idx1sz-1).fltarray,lonpoints,latpoints)';
profile3 = reshape(gribrec(idx1sz).fltarray,lonpoints,latpoints)';

    rnkval1 = MyBI([[profile1(rnklatoi,rnklonoi-1),profile1(rnklatoi,rnklonoi)];...
           [profile1(rnklatoi-1,rnklonoi-1),profile1(rnklatoi-1, rnklonoi)]],...
           rnkdlon,rnkdlat);
          rnkval2 = MyBI([[profile2(rnklatoi,rnklonoi-1),profile2(rnklatoi,rnklonoi)];...
           [profile2(rnklatoi-1,rnklonoi-1),profile2(rnklatoi-1, rnklonoi)]],...
           rnkdlon,rnkdlat);
          rnkval3 = MyBI([[profile3(rnklatoi,rnklonoi-1),profile3(rnklatoi,rnklonoi)];...
           [profile3(rnklatoi-1,rnklonoi-1),profile3(rnklatoi-1, rnklonoi)]],...
           rnkdlon,rnkdlat);

          vbgval1 = MyBI([[profile1(vbglatoi,vbglonoi-1),profile1(vbglatoi,vbglonoi)];...
           [profile1(vbglatoi-1,vbglonoi-1),profile1(vbglatoi-1, vbglonoi)]],...
           vbgdlon,vbgdlat);
          vbgval2 = MyBI([[profile2(vbglatoi,vbglonoi-1),profile2(vbglatoi,vbglonoi)];...
           [profile2(vbglatoi-1,vbglonoi-1),profile2(vbglatoi-1, vbglonoi)]],...
           vbgdlon,vbgdlat);
          vbgval3 = MyBI([[profile3(vbglatoi,vbglonoi-1),profile3(vbglatoi,vbglonoi)];...
           [profile3(vbglatoi-1,vbglonoi-1),profile3(vbglatoi-1, vbglonoi)]],...
           vbgdlon,vbgdlat);

    rnkgpm = 287*rnkval1*1/9.8*log(rnkpress/(rnkpress-165));
    vbggpm = 287*vbgval1*1/9.8*log(vbgpress/(vbgpress-165));

    rnkpro = [rnkpro;[rnkgpm,rnkval2,rnkval3]];
```

```
    vbgpro = [vbgpro;[vbggpm,vbgval2,vbgval3]];

clear profile1;

gribrec = read_grib(gribfn,fxgoi,'ScreenDiag',0);

for idx = 1:2:idx2sz
        profile2 = reshape(gribrec(idx).fltarray,lonpoints,latpoints)';
    profile3 = reshape(gribrec(idx+1).fltarray,lonpoints,latpoints)';

        rnkval1 = fxlvoi(ceil(idx/2)) + rnkinf.rnkele;
        rnkval2 = MyBI([[profile2(rnklatoi,rnklonoi-1),profile2(rnklatoi,rnklonoi)];...
            [profile2(rnklatoi-1,rnklonoi-1),profile2(rnklatoi-1, rnklonoi)]],...
          rnkdlon,rnkdlat);
        rnkval3 = MyBI([[profile3(rnklatoi,rnklonoi-1),profile3(rnklatoi,rnklonoi)];...
            [profile3(rnklatoi-1,rnklonoi-1),profile3(rnklatoi-1, rnklonoi)]],...
          rnkdlon,rnkdlat);

        vbgval1 = fxlvoi(ceil(idx/2)) + vbginf.vbgele;
        vbgval2 = MyBI([[profile2(vbglatoi,vbglonoi-1),profile2(vbglatoi,vbglonoi)];...
            [profile2(vbglatoi-1,vbglonoi-1),profile2(vbglatoi-1, vbglonoi)]],...
          vbgdlon,vbgdlat);
        vbgval3 = MyBI([[profile3(vbglatoi,vbglonoi-1),profile3(vbglatoi,vbglonoi)];...
            [profile3(vbglatoi-1,vbglonoi-1),profile3(vbglatoi-1, vbglonoi)]],...
          vbgdlon,vbgdlat);

    rnkpro = [rnkpro;[rnkval1,rnkval2,rnkval3]];
    vbgpro = [vbgpro;[vbgval1,vbgval2,vbgval3]];
end

%Sort by increasing altitude
rnkpro = myRevRow(sortrows(rnkpro));
vbgpro = myRevRow(sortrows(vbgpro));

%Remove the ground level altitude above Mean Sea Level from the
%geopotential height of the winds to get true layer heights above ground
%level.
rnkpro = [rnkpro(:,1)-rnkinf.rnkele,rnkpro(:,2),rnkpro(:,3)];
enddat = find(rnkpro(:,1) < 0);
if ~isempty(enddat)
    rnkpro = rnkpro(1:enddat-1,:);
end

vbgpro = [vbgpro(:,1)-vbginf.vbgele,vbgpro(:,2),vbgpro(:,3)];
enddat = find(vbgpro(:,1) < 0);
```

```
if ~isempty(enddat)
   vbgpro = vbgpro(1:enddat-1,:);
end

profile = struct('rnk',rnkpro,'vbg',vbgpro);

apreturn=profile;
%end gpmprocess

function ddate = filedate(fdate)
idx = findstr(fdate,'-');

dd = fdate(1:idx(1)-1);
mon = fdate(idx(1)+1:idx(2)-1);
yy = fdate(idx(2)+3:idx(2)+4);

switch mon
   case 'Jan'
      mm = 1;
   case 'Feb'
      mm = 2;
   case 'Mar'
      mm = 3;
   case 'Apr'
      mm = 4;
   case 'May'
      mm = 5;
   case 'Jun'
      mm = 6;
   case 'Jul'
      mm = 7;
   case 'Aug'
      mm = 8;
   case 'Sep'
      mm = 9;
   case 'Oct'
      mm = 10;
   case 'Nov'
      mm = 11;
   case 'Dec'
      mm = 12;
   otherwise
      mm = 0;
end
```

```
if (mm <= 9) mm = ['0' int2str(mm)]; end
if (dd <= 9) dd = ['0' int2str(dd)]; end

ddate = [mm dd yy];

%end filedate

function vdate = vtdate(recval)
fcsttime = recval.P1 + recval.hour;
add_days = floor(fcsttime/24);
hrsrem = mod(fcsttime,24);
dayinc = recval.day + add_days;

if (dayinc <= 9)
   dd = ['0' int2str(dayinc)];
else
   dd = int2str(dayinc);
end

if (recval.month <= 9)
   mm = ['0' int2str(recval.month)];
else
   mm = int2str(recval.month);
end

if (recval.year <= 9)
   yy = ['0' int2str(recval.year)];
else
   yy = int2str(recval.year);
end

if (hrsrem <= 9)
   hh = ['0' int2str(hrsrem)];
else
   hh = int2str(hrsrem);
end

vdate = [mm dd yy hh];
%end vtdate

function pg = makegp(griddesc);
%new method to use a cartesian grid overlay of a stereo projection to
%create the grid, used to genereate the data positions

%Ellipsoid parameters
```

```
e2wgs = 0.00669437999013;
ewgs = sqrt(e2wgs);
awgs = 6378137.0;
bwgs = 6356752.3;

%Information from the grib description utilized to generate the grid
%lattitude and longitude values for the grib grid
lonstep = griddesc.Nx;
latstep = griddesc.Ny;
positions = zeros(lonstep,latstep,2);
cartcoord = zeros(lonstep,latstep,2);
sdx = griddesc.Dx;
sdy = griddesc.Dy;

%set easting and southing in cartcoord for 1,1 position
natlat = deg2rad(griddesc.Lin1);
natlon = deg2rad(griddesc.Lov);
latstart = deg2rad(griddesc.La1);
lonstart = deg2rad(griddesc.Lo1);

mzero = cos(natlat)/sqrt(1-e2wgs*sin(natlat)^2);
tzero = tan(pi/4 - natlat/2)/((1-ewgs*sin(natlat))/(1+ewgs*sin(natlat)))^(ewgs/2);
n = sin(natlat);
f = mzero/(n*tzero^n);
rzero = awgs*f*tzero^n;

t = tan(pi/4 - latstart/2)/((1-ewgs*sin(latstart))/(1+ewgs*sin(latstart)))^(ewgs/2);
r = awgs*f*t^n;
theta = n*(lonstart - natlon);

cartcoord(1,1,1) = r*sin(theta);%easting
cartcoord(1,1,2) = rzero - (r*cos(theta));%northing

%fill grids with Dx, Dy changes in easting and northing
for otridx = 1:latstep
   for inridx = 1:lonstep-1
      cartcoord(inridx+1,otridx,1) = cartcoord(inridx,otridx,1)+sdx;
      cartcoord(inridx+1,otridx,2) = cartcoord(inridx,otridx,2);
   end

   if (otridx < latstep)
      cartcoord(1,otridx+1,1) = cartcoord(1,otridx,1);
      cartcoord(1,otridx+1,2) = cartcoord(1,otridx,2)+sdy;
   end
end
```

```
%fill position with lon,lat pairs from cartcoords
rprim = sqrt(cartcoord(:,:,1).^2 + (rzero - cartcoord(:,:,2)).^2);
tprim = (rprim/(awgs*f)).^(1/n);
thetaprim = atan(cartcoord(:,:,1)./(rzero - cartcoord(:,:,2)));

positions(:,:,1)=90 - rad2deg(2*atan(tprim.*((1-ewgs*sin(natlat))/...
    (1+ewgs*sin(natlat))).^(ewgs/2)));
positions(:,:,2)=rad2deg(thetaprim/n) + griddesc.Lov;

pg = positions;
%end makegp

function llindex = gridwalk(inposmat,lonoi,latoi)
%Function to step through the grid to find the upper right cell value that
%corresponds to the first set of values that exceed the input latitude and
%longitude values.
[nx,ny,dump] = size(inposmat);
lonidx = 1;
latidx = 1;

errmin = round(log(nx*ny)/2);

for indx = 1:errmin
        latrange = [inposmat(lonidx,:,1),latoi];
        latrange = sort(latrange);
        latidx = find(latrange==latoi)-1;

        lonrange = [inposmat(:,latidx,2)',lonoi];
        lonrange = sort(lonrange);
        lonidx = find(lonrange==lonoi)-1;
end

llindex = [lonidx+1,latidx+1];
%end gridwalk

function bival=MyBI(imat,posx,posy)
a = (imat(2,1) - imat(1,1))*posy + imat(1,1); %left value interpolation
b = (imat(2,2) - imat(1,2))*posy + imat(1,2); %right value interpolation

bival = (a - b)*posx + b; %Value interpolation
%end MyBI

function revidx = myRevRow(imat)
%Function to reverse row sort order.
```

```
idx = length(imat);

for i=1:idx
   tm(i,:)=imat(idx+1-i,:);
end

revidx = tm;
%end myRevRow
```

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Morris Driels
   Naval Postgraduate School
   Monterey, California